IMAGE CODING USING WAVELETS, INTERVAL WAVELETS AND MULTI-LAYERED WEDGELETS

BY

LEE WEI SIONG

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2006

Acknowledgments

I would like to thank my supervisor A/Prof. Ashraf A. Kassim for his guidance throughout the course of this research. I am especially appreciative of his help in reviewing various materials, and his belief in the work. I am also grateful to the very enthusiastic Dr. Wayne M. Lawton for his teachings and advices on the subject of wavelets. I would like to extend my appreciation to Dr. K. R. Rao and Dr. Piet van der Putten, for their generosity of advices, knowledge and experience. Also, to Mr. Francis Hoon for his assistance in many ways during my years in the laboratory.

Certainly, to these wonderful comrades, with whom I have shared many wonderful discussions over the coffee table: Seetoh Cheewah, Loke Kum Loong, Feng Wei, Sebastien Benoit, Yap Wee Hau, Yew Chor Wei, Saravana Kumarsamy, Teo Swee Ann and most certainly to Aunt May.

And finally, this work is dedicated to my wife Serena, a cheerful life companion who made it all possible with her wisdom and support.

Table of Contents

Summary

An image or signal can be represented by different bases that include sinusoids and wavelets. The question, in the context of compression, is which of these bases can give efficient and stable representations. The *wavelet transform* has played an important role in recent advances in image compression, overshadowing its predecessor – the cosine transform. In this thesis, we investigate issues regarding coding image edges, which are perceptually important to human vision. Primarily, our work focuses on the design and proposals of new bases, their corresponding analyzing techniques suitable for an embedded perceptual image coding framework.

The first contribution of the thesis is the exploration of the use of wavelets in image coding and the proposal of the *color set partitioning in hierarchical trees* (CSPIHT) algorithm for embedded color image encoding. Compared to various existing color coding solutions, the CSPIHT algorithm is able to achieve comparable or better performance than other state-of-the-art techniques despite its simplicity.

In the context of signal transform coding, we examined various 1D solutions which have been used to treat finite signal analysis. The study leads to the design of several new wavelets on the interval. To find a efficient representation of edges, we investigate the influence of singularities on wavelet coefficients in their vicinity and propose a new expansion using *interval wavelets* to provide an efficient representation of piecewise smooth signals. The main property of interval wavelet expansion is that, it can efficiently encode signal singularities, which usually carry visually important and meaningful information. Several new algorithms are also introduced to extend the new expansion to 2D images. Experiment shows that our proposed new compression technique can outperform JPEG2000 in terms of visual quality.

Finally, we study a particular novel analysis technique using objects called *wedgelets* that can be used to approximate 2D piecewise constant segments. In our review of wedgelet analysis for image coding, several limitations and inefficiency are observed in regards to approximation for image junctions, corners and ridge-like features. To these

problems, we introduce a *multi-layered wedgelet* technique as the solution. Additionally, a new object called the *erasing wedgelet* is used improve the robustness of wedgelet analysis for image approximation. Our proposed hybrid multi-scale wavelet-wedgelet image coding scheme is able to preserve macro features well enough to facilitate visual interpretation at very low bit-rate environments. In terms of visual quality, it is shown that wedgelet representations can also outperform JPEG2000.

List of Tables

List of Figures

List of Symbols

	$\langle f, g \rangle$	Inner product
	$\|\|f\|\|$	Norm
	$\lfloor x \rfloor$	Largest interger $n \leq x$
	$\lceil x \rceil$	Smallest interger $n \geq x$
Sets	$\mathbb{Z}$	Integers
	$\mathbb{N}$	Positive integers including 0
	$\mathbb{R}$	Real numbers
	$\mathbb{R}^+$	Positive real numbers
Signals	$f(t)$	Continuous time signal
	$f[n]$	Discrete signal
	$\mathbf{1}_{[a,b]}$	Indicator function which is 1 in $[a, b]$ and 0 outside.
Spaces	$\mathbf{L}^2(\mathbb{R})$	Finite energy functions
	$\mathbf{l}^2(\mathbb{Z})$	Finite energy discrete signals
	$\mathbf{C}^p$	p times continuously differentiable functions
	$\mathbf{C}^\infty$	Infinitely continuously differentiable functions
	$\mathbf{U} \oplus \mathbf{V}$	Direct sum of two vector spaces $\mathbf{U}$ and $\mathbf{V}$
	$\mathbf{U} \otimes \mathbf{V}$	Tensor product of two vector spaces $\mathbf{U}$ and $\mathbf{V}$
Operators	$f^{(p)}(t)$	Derivative $\frac{d^p f(t)}{dt^p}$ of order p
	$f \star g$	Convolution
Transforms	$\hat{f}(\omega)$	Fourier transform
	$\hat{f}[k]$	Discrete Fourier transform
	$Wf(u, s)$	Wavelet transform
	$W^{\text{int}} f(u, s)$	Interval wavelet transform

Abbrevation

bpp	*Bits-Per-Pixel*
CEZW	*Color Embedded Zerotree Wavelet*
CIE	*Commision Internationale de l'Eclairage*
CIF	*Common Intermediate Format*
Codec	*Coder/ Decoder*
CSPIHT	*Color Set Partitioning In Hierarchical Trees*
DCT	*Discrete Cosine Transform*
DWT	*Discrete Wavelet Transform*
EZW	*Embedded Zerotree Wavelet*
HSI	*Hue-Saturation-Intensity*
HVP	*Human Visual Perception*
HVS	*Human Visual System*
JPEG	*Joint Photographic Experts Group*
KLT	*Karhunen Lòeve Transform*
LIP	*List of Insignificant Pixel*
LIS	*List of Insignificant Set*
LSP	*List of Significant Pixel*
LZW	*Lempel-Ziv-Welch*
MPEG	*Moving Picture Experts Group*
MLW	*Multi-Layered Wedgelet*
MR	*Multi-Resolution*
MSE	*Mean Square Error*
PCA	*Principal Component Analysis*
PSNR	*Peak Signal to Noise Ratio*
RDP	*Recursive Dyadic Partitions*
RGB	*Red-Green-Blue*
SAD	*Sum of Absolute Difference*
SPIHT	*Set Partitioning In Hierarchical Trees*
SOT	*Spatial Orientation Tree*
YUV	*Luminance-Chrominance color space*
YCbCr	*Luminance-Chrominance color space*

1 Introduction

To reduce knowledge to what can be described in mathematical equations is absurd. It excludes vast ranges of our experience of life. —Matthieu Ricard.

Data compression refers to techniques of reducing the number of symbols (or energy) that are used in conveying information, by relying on the fact that perceived information usually exhibits certain degree of predictability and redundancy. The origin of man's fascination for information coding could be traced in history to the end of Dark Ages and Archaic period (ca. 1100–475 B.C.) when formal writing systems[1] and alphabets are developed. The earliest modern example of coding or compression would be the Morse code [1], developed by Samuel Morse and Alfred Vail in 1838 for telegraphy. It is only in the late 1940s that modern work on data compression gained interests when the *Information Theory* was developed with Claude Shannon's landmark paper, *A Mathematical Theory of Communication* [2]. Shannon's pioneering work essentially defined the information Age and inspires numerous works in data transmission or storage. The *Shannon-Fano* codes [3], developed by Shannon and Robert Fano in 1949, is the first example of *lossless* statistical compressor that systematically assigns shorter codewords for more frequent occurring characters. The Shannon-Fano codes were soon

[1]The Linear A pictogram system and the Linear B phonogram system.

overshadowed by the optimally lossless *Huffman* coding [4], which is still popular in modern applications after more than 50 years. Development for lossless compression techniques continues well into the 1980s, accumulating a wealth of algorithms such as adaptive Huffman-coding, *Lempel-Ziv-Welch* (LZW) [5][6][7], run-length encoding, arithmetric coding [8] and the Burrows-Wheeler transform [9].

Lossy or irreversible data compression is a related but different discipline in the field with its own history of development that possibly originated from the idea of creating sounds using pure tones. Following the work by Joseph Fourier on heat conduction in 1822 [10][11], it became clear by the mid-1800s that any sufficiently smooth function could be decomposed into sums of sinusoids of different frequencies. Thus, the idea of audio compression by band-limiting the frequency components was conceived and employed in telephony and sound recording in the late 1800s. In the early days of television in 1950s, there were attempts to do similar kinds of compression for still and moving images in order to reduce the large bandwidth required by television broadcast. Due to technological limitations and mathematical difficulties, serious efforts in this direction were not made until digital storage and image processing became common in the 1970s. The breakthrough came in 1974 when Ahmed, Natarajan and Rao introduced the *discrete cosine transform* (DCT) [12].

It was in the late 1980s that the lossy image and video compression techniques began to gain wide interest in the research communities and the industries. Essentially, these techniques revolve around the idea of achieving compression through transform coding and quantization. International standards for still and moving image compression, called *Joint Photographic Experts Group* (JPEG, 1987) [13][14][15][16][17] and *Moving Pictures Experts Group* (MPEG, 1988) [18][19][20] respectively, were then developed using variants of DCT technology. Transform coding based on DCT has its limitations which motivate a search for new ideas. Eventually, the idea of wavelets was rediscovered and refined, which would lead to the technology behind JPEG2000 [21][22]. The ideas and principles behind wavelets could be attributed to works of Joseph Fourier (1807), Alfred Haar (1909)[23], Paul Levy (1930s)[24][25] [26], Jean Morlet and Alex Grossman (1984) [27]. In 1986 [28], the foundation for modern wavelets was laid when Stéphane

Mallat, collaborating with Yves Meyer, showed that wavelets are implicit in the process of multiresolution analysis. Thus the theory of multiresolution analysis for wavelets [29][30] was developed, making wavelet analysis much easier. Around 1988, Ingrid Daubechies constructed a family of compactly supported orthogonal wavelets [31][32], now commonly known as the *Daubechies Orthogonal Wavelets*. The feasibility of wavelet transform computation in applications was realized when Mallat introduced a fast orthogonal wavelet transform algorithm [30] using multiresolution spaces. The following decade witnessed many novel ideas using wavelets for image analysis, compression and denoising that showed promising experimental results. Soon, several image coding algorithms based on wavelets are developed and they outperformed the DCT-based JPEG easily. Classic examples are the *embedded zerotree wavelet* (EZW, 1993, [33]) and the *set partitioning in hierarchical trees* (SPIHT, 1996, [34]) coding schemes. Soon, work begun for a new standard (JPEG2000) using the EBCOT (*Embedded Block Coding with Optimized Truncation*, [35]) wavelet technology to replace JPEG. The JPEG2000 core specification was approved in December 2000 with extended features approved in October 2001 [21].

1.1 Beyond JPEG2000

As the industry and public begin to adopt the JPEG2000 standard in recent years, the research community has already begun looking beyond wavelets. Like the DCT, there are still limitations to what wavelets are able to achieve in image coding. Particularly, wavelets are not well adapted to singularities beyond one-dimension. The success of the wavelet transform lies mainly in its frequency-space localization property and its ability to efficiently characterize certain classes of signals with few transform coefficients. However, the presence of edges in images, which are perceptually important, creates problems for efficient wavelet representation [2]. A very interesting reference [36] noted by Donoho shows how far mathematics and engineering still have to go before coding images with anything close to the efficiency of the human visual system:

> *The human eyes accept input at over* 10 *megabits per second. That input is highly redundant. At the end of processing, we actually acquire information at a rate closer to* 20 *to* 40 *bits per second on the conscious level. That compression ratio, enormous by any standard, is*

[2] Without doubt, they are still better than sinusoidal-based representations like DCT.

certainly far beyond anything that cosine or wavelet representations could achieve.

Since the beginning of 2002, there has been several proposals for new bases in multiresolution spaces, such as *beamlets*, *wedgelets* [37], *bandelets* [38], *curvelets* [39], *contourlets* [40][41], *ridgelets* [42] etc., which are also collectively termed as the '$\mathcal{X}$-lets' [43]. The principal idea behind these variants is to yield better representation of image discontinuities by the inclusion of orientation property in the bases. These ideas are still in their infancy and there exist implementation and mathematical problems when their formulation is translated to the discrete domain. Nevertheless, a few early experiments with these $\mathcal{X}$-lets have demonstrated promising results, as with the case of wavelets decades ago.

There are other approaches to image compression. Instead of obtaining best approximation for a given rate, an alternative is to find good representation or impression. This is analogous to an artist's perspective. These ideas and motivations stem from the different interpretations of the words *information* and *meaning*. The academic definition of *information* actually tells us nothing about the usefulness of a message, sound, image, video etc. It is only by coincidence that they seems to do so. French philosopher Jean Baudrillard [44] proposed three interesting hypotheses for information as follows:

- *Information* produces *meaning*, but cannot make up for the brutal loss of *signification* in every domain.
- *Information* has nothing to do with *significance* but it is something else, an operational model of another order, outside *meaning* and of the circulation of *meaning*.
- There is rigorous and necessary *correlation* between the two, to the extent that *information* is directly neutralizing or destroying both *meaning* and *significance*.

The second hypothesis is essentially that of Shannon's:

> *the space of information is purely functional, and only provides as a technical medium which has no implication to the finality of meaning, and thus should also not be implicated in a value judgement*.

In short, the meaningfulness of a message, or image as in our context, cannot be measured by information; it can only be validated by an observer through his perception.

(a) Original (b) JPEG2000, 0.2bpp (c) JPEG2000, 0.1bpp

Figure 1.1: Example of wavelet artifacts seen in a typical cartoon image. Notice the increasingly observable artifacts such as *edge blurring*, *halos* and *color distortion*.

(a) Original image (b) JPEG2000 (c) Paintbrush method

Figure 1.2: Example of different approaches to image reconstruction. JPEG2000 and its predecessor attempts to find the best approximation. Paintbrush technique models the best representation that a human observer would assume.

Unlike audio compression where there is a successful standard psycho-acoustic model for the ear, an equivalent model for the *human visual system* (HVS) is still lacking. Inevitably, the forthcoming generation of image and video compression techniques would place great emphasis on human visual perception, not only physiologically but also at the psychological level.

1.2 Graphic Visualization and Perceptual Ordering

In a typical image and video coding-decoding scenario, the viewer usually has no prior knowledge or preview of the original source. Hence, with limited coding resources, the approach to approximate the image in its wholeness is not only unnecessary

but will also definitely lead to intolerable visual distortions, such as *ringing*, *blocking* and *'mosquito'* (see figure 1.1). Hence, we suggest achieving image compression by eliminating irrelevant features at a given rate in order to give a best possible visual representation. This approach is similar to an artist making a sketch given time and resource constraint, in which not all scene contents are portrayed nor reproduced accurately. Nevertheless, the drawing would be a sufficiently good representation that allows reasonable visual interpretation. In fact, this approach has been attempted by a few, such as [45][46] where templates of brush strokes of different scales, orientations and colors are used to *render* an image (see figure 1.2). This technique shares the same insight with artists in drawing and painting. Unfortunately, such an unconventional approach lacks a formal framework for proper mathematical analysis and thus, has been largely neglected by the community.

What would be considered a good representation? To answer that, it is necessary to first ask how people see images. Believing that colors should mix in the eye and not in the artists palette, French painter Georges Seurat pioneered the *Pointillism* technique in the 1880s– painting only with dots of primary colors. As visionary as he may be, a century later, propelled by technology, we are now all *pointillists*. Today, images are displayed and replicated by dots of light and pigments. From computer displays to printed materials, we perceive images as collection of fine dots. Even so, the retina of the mammalian visual system, is clearly a photon sampling device, having much in common with photo sensor arrays in a digital camera. It is only natural that we decompose and reconstruct images as an array of colored dots. Yet, we do not see in pixels. At the level of conscious awareness, what we see are faces, trees, cars, birds, etc. Processing images as arrays of pixels has both its strength and weakness. On the one hand, the structural simplicity of pixels facilitates many useful image processing techniques such as filtering, sharpening, equalization and color enhancement. On the other hand, pixels are oblivious to what they are portraying– absolutely ignorant of the patterns and structures they form.

An important principle employed in human vision is *unconscious inference* [48][49]. Hidden assumptions are used with retinal images to determine the perception of a

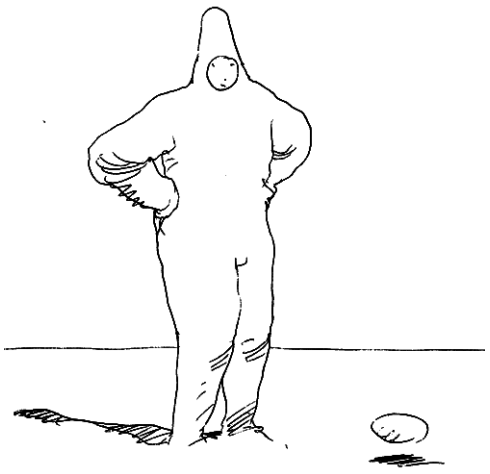


Figure 1.3: A simple line drawing (original sketch by Jean Giraud).

scene. People are often not aware of such visual inferences they are making [50]. The likelihood principle states that people accept the view that has the highest probability of occurrence. The heuristic principle states that people make inferences about the most likely environmental conditions that could produce the image. Thus, image artifacts produced in current coding technology such as the JPEG2000 shown in figure 1.1(b), 1.1(c) and 1.2(b), can be visually 'annoying' and unacceptable because they are not 'natural' to our visual experience. Moreover, these artifacts will prompt the observer to judge the image as 'poor' since they hinder information processing and interpretation. We propose that a good representation should be free of distortions that are unfamiliar and unnatural to the observer's visual knowledge base.

Surely some information must be lost, but in what manner should this be done? Consider an art painting, which is often an impression by an artist on a subject rather than an exact representation the reality. Not all details in a scene are painted but only those that are deemed important and relevant to the artist. A very simple cartoon image (see figure 1.3) can be a good representation of reality with minimal details as long as correct and sensible information is being conveyed.

In the context of generating good representations, the next issue is to know which visual details are important— both absolutely and relatively. In his classic book [51][47] for graphic designers, Bertin introduces the concept of retinal variables in relation to

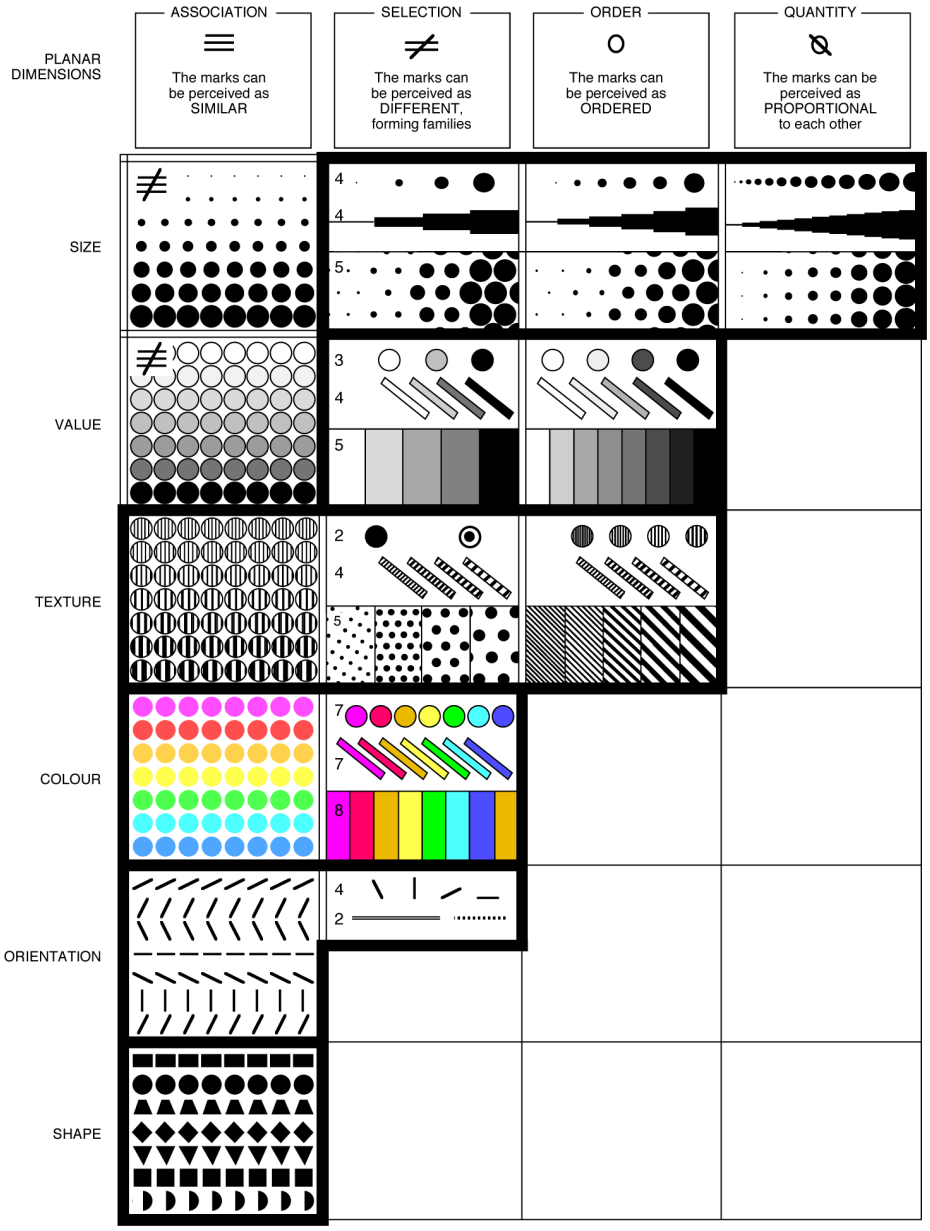


Figure 1.4: Illustration of the levels of organization of each of Bertin's visual variables: *spatial planar dimensions*, *size*, *value*, *texture*, *color*, *orientation* and *shape*. The numbers shown on the left of the Selection column indicate the recommended number of levels to support selective perception by each implantation (point, line and area, respectively). (reproduced from [47], page 96)

the psychology of human perception that can be manipulated by designers to optimize visualized data. The retinal variables are *planar dimensions*, *size*, *value* or *intensity*, *texture*, *color*, *orientation* and *shape*, which are essentially the factors that are psychologically important in human vision. Figure 1.4 shows that the levels of organization of these retinal variables, like the levels of organization of information, are either *qualitative* (involving associative or selective perception), *ordered*, or *quantitative* and variables are either *associative* ($\equiv$), *dissociative* ($\not\equiv$), *selective* ($\neq$), *ordered* ($\circ$), or *quantitative* ($\oslash$). The evidence of retinal variable classification and ordering has been observed by neuroscientists. In the 1960s, Hubel and Wiesel [52][53][54][55] recorded the response of individual V1[3] neurons. These experiments revealed that the V1 cortex seems to classify features according to their position, orientation and angular size. Clearly, visual processing should be based on a distinct visual hierarchy in which certain perceptual elements have priority. The groups of retinal variables, which are either selective or ordered, are of particular interest to us. Selectivity allows us to make distinctions between different spatial regions. Ordered variables allows us to rank image elements

There is another important aspect to the psychology of human vision, i.e., the tendency to categorize the perceived vision as *foreground* and *background*. Familiarity or past experience plays a central role in assisting us to make that distinction. In fact, *figure-ground segregation* is one of the principles in Gestalt theory [56][57] for perception. The object boundary is one of the most important visual cue in perception [58], which allows the ability to distinguish objects from the background. But what is a boundary? Referring to Bertin's table, the variables that are selective — *size*, *value*, *texture* and *color* — allows the perception of a boundary's presence. Thus we often see edge detection in image processing being performed by measuring image intensity and hue variations. In more advanced techniques, textures and motion fields are used to obtain boundaries. However, not all boundaries detected by such techniques necessarily lead to meaningful perception of objects.

Bertin suggested in [51] that *size* is the most important retinal variable, that larger objects are preferably being focused on than smaller ones. Support for this claim can

[3] also known as primary visual cortex or the striate cortex.

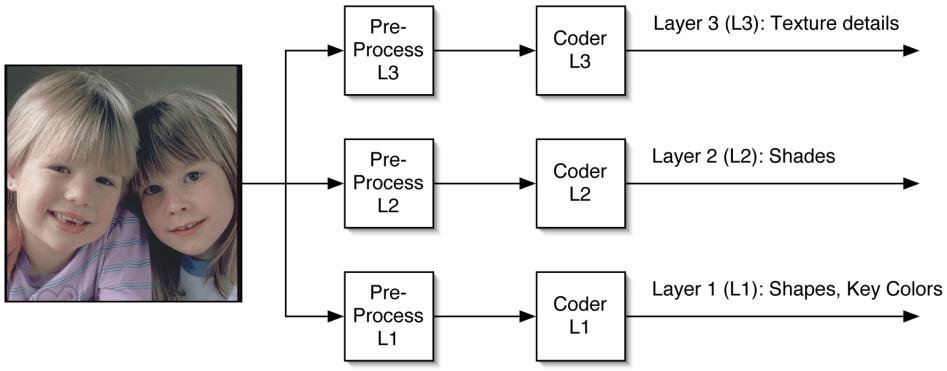


Figure 1.5: The encoding framework in an embedded perceptual system.

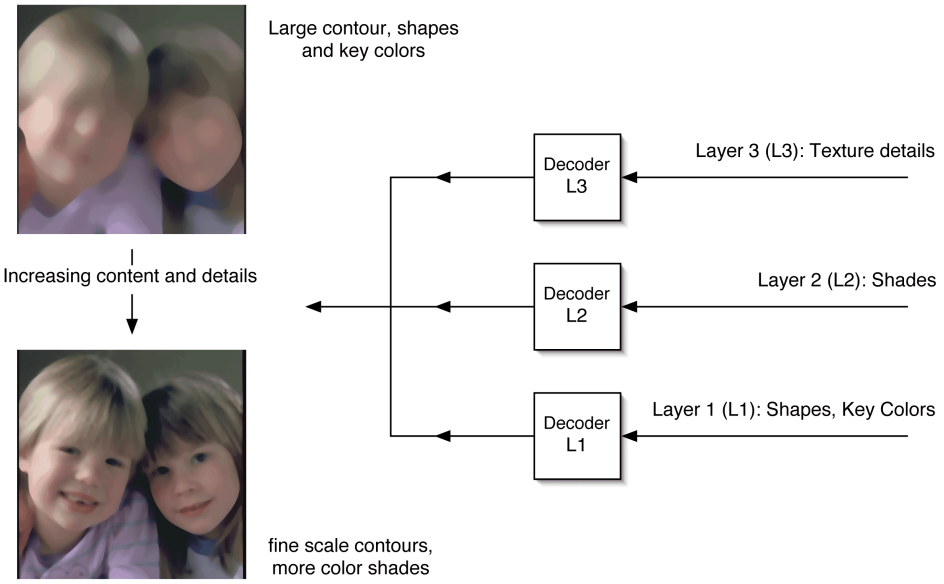


Figure 1.6: The decoding framework in an embedded perceptual system.

be found in [59][60]. The idea of multiresolution in wavelet theory essentially re-emphasized the importance of scales in analysis. This concept corresponds to ordering sizes and textures in Bertin's table.

From the discussion so far, there are two key observations:

- Boundaries are very important in human vision as information cues.
- The ordering of visual data is the natural way of prioritizing and processing information in a human vision system.

These observations lead us to propose the image coding framework[4] shown in figures 1.5

[4]a framework provides a basic set of parts which may be used to develop and build further parts, and

and 1.6. In our proposed framework, an image is preprocessed into different perceptual layers and each layer will have its own encoder adapted to the nature of data in the corresponding layer. In figures 1.5 and 1.6, we have layers in decreasing perceptual order of importance

- **L1**. This can be called the *primitive* layer since it contains features that are essential to low-level vision [61][62] [63][64]. It provides essential information that can help in object detections such as key colors and boundaries. Reconstruction using only data from L1 should give a cartoon-like image.

- **L2**. Intensity and hue variation information are found in this layer, which provides a perception of *depth* to the image from L1. Thus piecewise smooth signal is expected on this layer.

- **L3**. This layer consist of textures details which gives *richness* to images constructed from L2 and L1. Texture is a significant coding issue to all current transform coding methods including wavelets. In our opinion, textures should be rendered by synthesis methods, instead of being coded. However, the discussion of this topic is entirely outside the scope of this thesis and hence will not be dealt with.

On the decoding side, depending on the bandwidth or bit resources, the image is reconstructed using L1, L2 and L3 data in that order, and the final image is obtained by superimposing the reconstructions from each layer. For example, at very low bit-rate, the image will only be reconstructed from information in L1, thus resulting in a cartoon-like image. With more bit resources, information from L2 channel can be decoded and the image can be refined with more shades and finer details. Hence, this proposed image coding framework is *perceptually scalable*.

1.3 Proposal and Objectives

Our work revolves around coding issues for image data from L1 and L2 layers, In relation to the framework proposed in figure 1.5 and 1.6, the objective of our research is to design analysis tools and coding techniques for L1 and L2 layers using wavelets and

is therefore capable of producing systems. The distinction is that a framework gives a supportive structure for the future development of new parts whereas a system does not.

new bases beyond wavelets. These problems are generally related to the field of signal approximations, singularity and edge detection, and filter design.

1.3.1 Overview of thesis

We briefly describe the contents of the chapters that follow in the remainder of this thesis:

- *Chapter 2: Wavelet Preliminaries*— We review the theory of wavelets, their properties, construction, analysis and synthesis techniques. This chapter lays the foundation of our discussion for the subsequent chapters in wavelet filter construction and wavelet representations in image compression.

- *Chapter 3: Wavelets and Image Embedded Coding*— We discuss the ideas of embedded coding and introduce several classical algorithms. Different color variants of embedded image coding are reviewed and we propose our embedded coding solution for color images. This is also where we observe some of the limitations of current wavelet methods for image compression.

- *Chapter 4: Analysis and Synthesis of Finite Signals*— We look at how finite data are being handled in signal processing, particularly in the wavelet transform. We design and introduce two new boundary filters which allows robust analysis of any arbitrary length signal.

- *Chapter 5: Signal Singularities: Detection, Analysis and Synthesis*— We investigate how wavelet coefficients behave in the presence of singularities and how reconstructed signal can be adversely affected by quantization errors. A new approximation by interval wavelets expansion is presented that minimize errors in the vicinity of singularities.

- *Chapter 6: Perceptual Image Compression I*— We present decomposition and reconstruction algorithms that extend interval wavelet analysis to 2D images.

- *Chapter 7: Perceptual Image Compression II*— We propose a new multi-layered wedgelet technique to improve the wedgelet approximation of images. A hybrid multiscale wavelet-wedgelet image coding scheme is also presented to code

animation cel-images.

1.3.2 Contributions Summary

We summarize below the key contributions of this work:

- A novel embedded color image coding using wavelets.
- New design of two families of orthogonal wavelets on the interval.
- General algorithm for computing orthogonal boundary filter coefficients and the corresponding pre/post-conditioning matrices for various families and vanishing moments.
- Improved 1D signal approximation using families of new interval wavelets.
- Algorithm for 2D image approximation using interval wavelets.
- Improved wedgelet analysis using multi-layered wedgelets.
- Hybrid wavelet-wedgelet image coding framework.

2 Wavelet Preliminaries

If we knew what it was we were doing, it would not be called research, would it? —Albert Einstein (1879-1955).

The wavelet transform or wavelet analysis is probably the most recent solution to overcome the shortcomings of the Fourier transform. The fundamental idea behind wavelets is to analyze according to scale. A wavelet, as the name suggests, is a small oscillatory function with its energy localized in frequency-space. Wavelet transform allows simultaneous frequency and time analysis so that it is an appropriate tool to analyze transient, non-stationary, or time-varying phenomena of different sizes and scales. This concept of localized frequency events has always been an obvious natural perception, especially to the auditory system. Generally, wavelet coefficients can better represent transient behavior, such as *staccati* in music. In other words, some elements of the signal that are transient can be represented by a smaller amount of information than the case where sinusoidal-based transforms are used.

In this chapter, we introduce the wavelet theory. Some of the essential concepts are the multiresolution analysis, the imposed conditions for wavelet function design and their properties. Understanding of these concepts will facilitate discussions of the

works in subsequent chapters, where we discuss the construction of boundary filters and wavelet edge detections.

2.1 Multiresolution Analysis

The idea of *multiresolution* (MR) approximation is important and fundamental to the wavelet theory. By adapting the signal resolution, we can process only the relevant details for particular task. Burt *et al.* [65] first proposed in computer vision to utilize a multiresolution pyramid for approximating images at various scales and resolutions. From literature it is not always clear what is meant by small and large scales, and for clarity, we define these as follows:

> the large scale gives a wider view scope, while the small scale shows the details.

Thus, going from large scale to small scale is, in this context, equivalent to zooming. In this section, we formalize the MR approximations, which will be the basis for construction of orthogonal wavelets.

The approximation f_j of a function f at resolution 2^{-j}, or scale j, is defined as an orthogonal projection on a certain space $\mathbf{V}_j \subset \mathbf{L^2}(\mathbb{R})$ such that the measure $\|f - f_j\|$ is minimized. The space $\mathbf{V}_j$ contains all possible approximations at resolution 2^{-j}. The following MR definition was introduced and formalized by Mallat [30][66], Daubechies [31] and Meyer [29][67][68], in which the mathematical properties required of the multiresolution spaces are specified.

Definition 2.1.1. *(Multiresolutions) A sequence* $\{\mathbf{V}_j\}_{j\in\mathbb{Z}}$ *of closed subspaces of* $\mathbf{L^2}(\mathbb{R})$ *is a multiresolution approximation if the following* 6 *properties are satisfied.*

$$\forall (j,k) \in \mathbb{Z}^2, \quad f(t) \in \mathbf{V}_j \Leftrightarrow f(t - 2^j k) \in \mathbf{V}_j, \tag{2.1}$$

$$\forall j \in \mathbb{Z}, \quad \mathbf{V}_{j+1} \subset \mathbf{V}_j, \tag{2.2}$$

$$\forall j \in \mathbb{Z}, \quad f(t) \in \mathbf{V}_j \Leftrightarrow f(t/2) \in \mathbf{V}_{j+1}, \tag{2.3}$$

$$\lim_{j\to+\infty} \mathbf{V}_j = \bigcap_{j=-\infty}^{+\infty} \mathbf{V}_j = \{0\}, \tag{2.4}$$

$$\lim_{j\to-\infty} \mathbf{V}_j = Closure\left(\bigcup_{j=-\infty}^{+\infty} \mathbf{V}_j\right) = \mathbf{L^2}(\mathbb{R}), \tag{2.5}$$

and there exists θ such that $\{\theta(t-n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of $\mathbf{V}_0$.

Property (2.1) means that $\mathbf{V}_j$ is invariant to translation proportion to scale 2^j. Property (2.2) is a causality property implying that an approximation at scale 2^j contains all necessary information to compute an approximation at coarser scale 2^{j+1}. Naturally, this leads to nested spaces:

$$\ldots \subset \mathbf{V}_2 \subset \mathbf{V}_1 \subset \mathbf{V}_0 \subset \mathbf{V}_{-1} \subset \mathbf{V}_{-2} \subset \ldots \tag{2.6}$$

Property (2.3) defines the approximation at coarser scales by dilation. Property (2.4) implies that as the scale tends to $+\infty$, i.e., the resolution 2^{-j} tends to 0, we lose all details of f:

$$\lim_{j\to+\infty} \|P_{\mathbf{V}_j} f\| = 0. \tag{2.7}$$

On the other hand, property (2.5) denotes that the approximation converges to the original signal when the resolution tends to $+\infty$:

$$\lim_{j\to-\infty} \|f - P_{\mathbf{V}_j} f\| = 0. \tag{2.8}$$

For the last property, the existence of a Riesz basis $\{\theta(t-n)\}_{n\in\mathbb{Z}}$ for $\mathbf{V}_0$ provides a numerically stable expansion of signals $f \in \mathbf{V}_0$ over the basis. In other words, there exists bounds $A > 0$ and $B > 0$ such that

$$f(t) = \sum_{n=-\infty}^{+\infty} a_n \theta(t-n) \tag{2.9}$$

with

$$A\|f\|^2 \leq \sum_{n=-\infty}^{\infty} |a_n|^2 \leq B\|f\|^2. \tag{2.10}$$

It can be easily verified that the family $\{2^{-j/2}\theta(2^{-j}t-n)\}_{n\in\mathbb{Z}}$ is a Riesz basis of $\mathbf{V}_j$ with the same bounds A and B for all scales j. When the Riesz basis is an orthogonal basis, the multiresolution approximation is orthogonal, and the base atom is called a scaling

function ϕ.

Before we proceed to discuss the bases that constitute the MR spaces, we take a look at the constraints on our signal: *the signal to be analyzed must have finite energy*. When the signal has infinite energy, it is impossible to cover its frequency spectrum and time duration with wavelets. Formally, the constraint is stated as

$$\int_{-\infty}^{+\infty} |f(t)|^2 dt < \infty. \tag{2.11}$$

Equivalently speaking, the $\mathbf{L}^2$-norm of the signal must be finite, thus existing in the Hilbert space.

2.1.1 Scaling Functions ϕ

The wavelet transform is based on the concept of multiresolution analysis, which partitions $\mathbf{L}^2(\mathbb{R})$ into a nesting of spanned spaces (see eqn. 2.6). Obviously, when we perform wavelet analysis, we cannot proceed indefinitely to larger scales. The analysis has to terminate at some space $\mathbf{V}_{j_0}$. Thus the analyzing function to 'terminate' the process is a scaling function ϕ_{j_0} [30] that spans $\mathbf{V}_{j_0}$. Similarly on the other end of the infinite nested space, prior to analyzing a function $f \in \mathbf{L}^2(\mathbb{R})$, we need to project it onto some initial space, say $\mathbf{V}_0$:

$$P_{\mathbf{V}_0} f = f_0 \in \mathbf{V}_0.$$

Subsequently, we need to calculate $P_{\mathbf{V}_j} f$ for other scales j, i.e., approximations of f at different scales or resolutions. To compute these projections, we must find the family of orthonormal basis ϕ_j of $\mathbf{V}_j$ which will characterize the entire multiresolution approximation. By defining a set of scaling functions by integer translates of the basic scaling function ϕ,

$$\phi_{0,k}(t) = \phi(t-k), \quad k \in \mathbb{Z} \tag{2.12}$$

then the subspace $\mathbf{V}_0$ of $\mathbf{L}^2(\mathbb{R})$ can be defined as

$$\mathbf{V}_0 = \overline{\text{span}\{\phi_{0,k}(t) : k \in \mathbf{Z}\}}. \tag{2.13}$$

For ϕ to be qualified as orthonormal basis of $\mathbf{V}_0$, it has to satisfy the following:

- The 0^{th} moment of the scaling function cannot vanish:

$$\int_{-\infty}^{\infty} \phi(t)dt = 1. \tag{2.14}$$

  or equivalently, $\hat{\phi}(0) = 1$.

- It has unit energy:

$$\|\phi(t)\|^2 = \int_{-\infty}^{\infty} |\phi(t)|^2 dt = 1. \tag{2.15}$$

- It is orthogonal to its integer translates:

$$\langle \phi(t), \phi(t-n) \rangle = \delta(n). \tag{2.16}$$

From the basic scaling function ϕ, a family of functions can be generated by dyadic scaling and translation,

$$\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k), \quad (j,k) \in \mathbb{Z}^2, \tag{2.17}$$

spanning the subspaces

$$\mathbf{V}_j = \overline{\text{span}\{\phi_k(2^j t) : k \in \mathbf{Z}\}} = \overline{\text{span}\{\phi_{j,k}(t) : k \in \mathbf{Z}\}} \tag{2.18}$$

Eqn. (2.17) automatically satisfies $\phi_{j,k}$ as Riesz basis of $\mathbf{V}_j$ with tight bounds and the rest of the multiresolution conditions (2.1) thru (2.5) are fulfilled. The subspace nesting in (2.6) implies that if $\phi \in \mathbf{V}_0$, we have $\phi \in \mathbf{V}_{-1}$, which is the space spanned by $\phi(2t)$ given by the causality property (2.2). Thus $\phi(t)$ can be expressed as a linear combination of $\phi(2t)$:

$$\phi(t) = \sqrt{2} \sum_n h_n \phi(2t - n), \tag{2.19}$$

where

$$h_n = \langle \phi(t), \phi(2t - n) \rangle. \tag{2.20}$$

Eqn. (2.19) is known by many names: *refinement*, *dilation*, *scaling*, or *two-scale difference* equation [69][70][71]; and h is known as the scaling filter. We can derive some properties of the coefficients of h directly from the dilation equation. Integrating both sides of eqn. (2.19), we obtain

$$\sum_k h_k = \sqrt{2}. \tag{2.21}$$

On the other hand, if we multiply both sides of eqn. (2.19) with $\phi(t-l)$ and integrate, we have

$$\int \phi(t)\phi(t-l)dt = 2\sum_k \sum_{k'} h_k h_{k'} \int \phi(2t-k')\phi(2t-2l-k)dt = \sum_k h_k h_{k+2l}. \tag{2.22}$$

Using orthogonality of ϕ in eqn. (2.16), we have a double-shift orthogonality condition on h:

$$\sum_k h_k h_{k+2l} = \delta_{0,l}. \tag{2.23}$$

These properties of h are important for the dilation equation to have a solution, i.e., it converges to a scaling function ϕ. The following theorem 2.1.2 formally gives the necessary and sufficient conditions on h.

Theorem 2.1.2 (Mallat, Meyer)**.** *Let $\phi \in \mathbf{L}^2(\mathbb{R})$ be an integrable scaling function. The Fourier series of $h_n = \langle 2^{-\frac{1}{2}}\phi(t/2), \phi(t-n)\rangle$ satifies*

$$\forall \omega \in \mathbb{R}, \quad |\hat{h}(\omega)|^2 + |\hat{h}(\omega+\pi)|^2 = 2, \tag{2.24}$$

and

$$\hat{h}(0) = \sqrt{2}. \tag{2.25}$$

Conversely, if $\hat{h}(\omega)$ is 2π periodic and continuously differentiable in a neighborhood of $\omega = 0$, if it also satisfies eqn. (2.24) and (2.25) and if

$$\inf_{\omega \in [-\frac{\pi}{2}, \frac{\pi}{2}]} |\hat{h}(\omega)| > 0 \tag{2.26}$$

then

$$\hat{\phi}(\omega) = \prod_{p=1}^{+\infty} \frac{\hat{h}(2^{-p}\omega)}{\sqrt{2}} \tag{2.27}$$

is the Fourier transform of a scaling function $\phi \in \mathbf{L}^2(\mathbb{R})$.

Note that eqn. (2.24) is equivalent to eqn. (2.23). Also, eqn. (2.25) and eqn. (2.21) further tells us that h is necessary a low-pass filter. This make sense since as the scale $j \to +\infty$, we lost all the details, and the space $\mathbf{V}_j$ contains only coarsely approximated functions.

2.1.2 Wavelet functions ψ

Now we examine the spaces the bridge the difference between $\mathbf{V}_j$ and $\mathbf{V}_{j+1}$. Let $\mathbf{W}_j$ be the orthogonal complement of $\mathbf{V}_j$ in $\mathbf{V}_{j+1}$:

$$\mathbf{V}_{j-1} = \mathbf{V}_j \oplus \mathbf{W}_j. \tag{2.28}$$

Iterating gives

$$\mathbf{V}_{j+1} = \bigoplus_{k=-\infty}^{j} \mathbf{W}_k, \tag{2.29}$$

and letting $j \to +\infty$, we have

$$\mathbf{L}^2(\mathbb{R}) = \bigoplus_{k=-\infty}^{+\infty} \mathbf{W}_k. \tag{2.30}$$

This says that we can decompose $\mathbf{L}^2(\mathbb{R})$ into orthogonal subspaces, each containing information about details at a given resolution. So the collection of the bases of $\{\mathbf{W}_j\}$ serves as a basis for $\mathbf{L}^2$. It turns out that for a multiresolution analysis, the detail space $\mathbf{W}_j$ has a set of orthonormal basis $\psi_{j,k}$ called *wavelets*, where

$$\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k). \tag{2.31}$$

Each wavelet $\psi_{j,k}$ is generated by translation and dilation from a single function ψ, which is referred to as the *mother wavelet*.

Since the wavelets ψ reside in the space spanned by the next finer scaling function,

i.e., $\mathbf{W}_0 \subset \mathbf{V}_{-1}$, we can write similar dilation relations for ψ:

$$\psi(t) = \sqrt{2} \sum_n g_n \phi(2t - n) \tag{2.32}$$

where

$$g_n = \sqrt{2}\langle \psi(t), \phi(2t - n) \rangle. \tag{2.33}$$

Eqn. (2.32) is called the *wavelet equation* which relates the mother wavelet ψ to the scaling function ϕ at the next finer scale. The following theorem formally summarizes our discussion on ψ.

Theorem 2.1.3. *Let ϕ be a scaling function and h the corresponding conjugate mirror filter. Let ψ be the function whose Fourier transform is*

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2}} \hat{g}(\frac{\omega}{2}) \hat{\phi}(\frac{\omega}{2}), \tag{2.34}$$

with

$$\hat{g}(\omega) = e^{-i\omega} \hat{h}^*(\omega + \pi). \tag{2.35}$$

Denote

$$\psi_{j,n}(t) = 2^{-j/2} \psi(2^{-j}t - n). \tag{2.36}$$

Then, for any scale 2^j, $\{\psi_{j,n}\}_{n \in \mathbb{Z}}$ is an orthonormal basis of $\mathbf{W}_j$. For all scales, $\{\psi_{j,n}\}_{(j,n) \in \mathbb{Z}^2}$ is an orthonormal basis of $\mathbf{L}^2(\mathbb{R})$.

Eqn. (2.34) is actually the Fourier transform of eqn. (2.32). From eqn. (2.35) and using eqn. (2.24) and (2.25), we obtain

$$\hat{g}(0) = 0 \quad \text{and} \quad \hat{g}(\pi) = -\sqrt{2}.$$

Thus g is necessarily a band-pass filter. By applying inverse Fourier transform to eqn. (2.35), we can get the coefficients of g in relation to the low-pass filter h as

$$g[n] = (-1)^n h[1 - n]. \tag{2.37}$$

2.2 Filter bank and Fast Wavelet Transform

So far we have discussed the nature of the analyzing scaling and wavelet functions. In this section, we will look at how a wavelet transform can be performed by what is called subband filtering in signal processing. We first define subband filtering and show that wavelet decomposition simply amounts to subband filtering with a pair of lowpass and highpass filters.

In signal processing, a signal is often decomposed or separated into different frequency bands or channels after which it can be coded and transmitted efficiently. This decomposition of signal into different frequency channels is called *subband filtering* and it is usually done using a collection of parallel filters and decimators called a *filter bank*. It usually consists of an *analysis* bank and *synthesis* bank, designed to separate an input signal into subbands and then to recombine these subbands. Since the signal is split into multiple subbands, there is an expansion and redundancy in the subband filtered data. Hence the decimators or downsamplers are necessary in the filter bank system

One important task of an analysis/synthesis system is the reconstruction of the input signal. The ideal case is *perfect reconstruction* (PR), where the output signal is the same as the input signal except for a delay and a scaling factor. A two channel filter bank is illustrated in figure 2.1. The reconstructed signal x_1 is obtained by filtering the upsampled (zeroes-interleaved) signals with a low-pass h_1 and high-pass g_1. An explicit expression for the reconstructed signal can be obtained in terms of the input:

$$x_1[n] = \sum_l \sum_m (\tilde{h}_{n-2l} h_{2l-n} + \tilde{g}_{n-2l} g_{2l-n}) x_0[m]. \tag{2.38}$$

Thus, to have perfect reconstruction, we must have

$$\sum_l (\tilde{h}_{n-2l} h_{2l-n} + \tilde{g}_{n-2l} g_{2l-n}) = \delta_{n,m}. \tag{2.39}$$

The following theorem due to Vetterli *et al.* [72] gives the conditions in frequency space for perfect reconstruction.

Theorem 2.2.1 (Vetterli)**.** *The filter bank performs an exact reconstruction for any input signal*

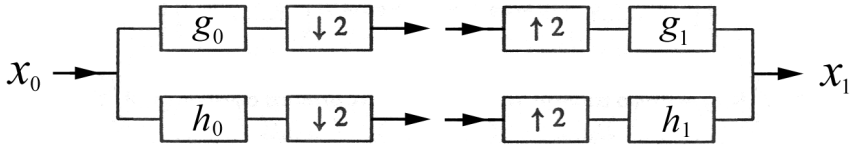


Figure 2.1: Two channel filter bank.

if and only if

$$\hat{h}^*(\omega+\pi)\hat{\bar{h}}(\omega)+\hat{g}^*(\omega+\pi)\hat{\bar{g}}(\omega)=0, \tag{2.40}$$

and

$$\hat{h}^*(\omega)\hat{\bar{h}}(\omega)+\hat{g}^*(\omega)\hat{\bar{g}}(\omega)=2. \tag{2.41}$$

Recalling eqn. (2.35), the choice formulation of g is not accidental but due to perfect reconstruction requirement which follows:

Theorem 2.2.2. *Perfect reconstruction filters satisfy*

$$\hat{h}^*(\omega)\hat{\bar{h}}(\omega)+\hat{h}^*(\omega+\pi)\hat{\bar{h}}(\omega+\pi)=2. \tag{2.42}$$

If the filters have a finite impulse response then there exists $a\in\mathbb{R}$ and $b\in\mathbb{Z}$ such that

$$\hat{g}=ae^{-j(2b+1)\omega}\hat{\bar{h}}^*(\omega+\pi) \tag{2.43}$$

and

$$\hat{\bar{g}}=a^{-1}e^{-j(2b+1)\omega}\hat{h}^*(\omega+\pi) \tag{2.44}$$

Both eqn. (2.43) and (2.44) are Fourier equivalents of eqn. (2.35). Nonetheless, a PR filter bank (or any pair of conjugate mirror filters) does not necessarily generate a wavelet system. Indeed, some attention has been paid to the stability of the decomposition and reconstruction schemes as the scale increases, that is when the number of filter bank cascades goes to the infinity. This is expressed by an additional condition in theorem 2.1.2 on the conjugate mirror filter h.

Now, given a set of functions ϕ and ψ that describe the multiresolutional spaces of

$\mathbf{L}^2(\mathbb{R})$, any function $f(t) \in \mathbf{L}^2(\mathbb{R})$ can then be written as

$$f(t) = \sum_k c_0[k]\phi_{0,k}(t) + \sum_{j=0}^{\infty}\sum_k d_j[k]\psi_{j,k}(t), \tag{2.45}$$

where c_0 and $\{d_j\}_{j\geq 0}$ are called the *discrete wavelet transform* (DWT) coefficients of the signal f. The first summation gives the coarse approximation of f, and the second term provides the details. For orthogonal ϕ and ψ, the coefficients can be derived as

$$c_j[k] = \langle f, \phi_{j,k}\rangle = 2^{-j/2}\int f(t)\phi(2^{-j} - k)dt \tag{2.46}$$

and

$$d_j(k) = \langle f, \psi_{j,k}\rangle = 2^{-j/2}\int f(t)\psi(2^{-j} - k)dt. \tag{2.47}$$

Rewriting the refinement eqn. (2.19) into general scaling and translation, we have

$$\phi(2^{-j} - k) = \sum_n h_n\phi(2^{-j+1})t - 2k - n). \tag{2.48}$$

Substituting eqn. (2.48) into eqn. (2.46), we obtain:

$$c_j[k] = \sum_n h_n \int f(t)2^{-(j-1)/2}\phi(2^{-j+1}t - 2k - n)dt. \tag{2.49}$$

Let $m = 2k + n$,

$$c_j[k] = \sum_m h[m - 2k]\int f(t)2^{-(j-1)/2}\phi(2^{-j+1}t - m)dt. \tag{2.50}$$

The scaling function inside the integral is at scale $j - 1$, thus eqn. (2.48) can be further simplified to

$$c_j[k] = \sum_m h[m - 2k]c_{j-1}[m]. \tag{2.51}$$

Similarly, the detail coefficients can be derived as

$$d_j[k] = \sum_m g[m - 2k]c_{j-1}[m]. \tag{2.52}$$

Both eqns. (2.51) and (2.52) shows that the scaling and wavelet coefficients at coarser scale j can be obtained by convolving the coefficients at scale $j-1$ by the filters h and g followed by decimation. These equations can in fact be easily implemented by a filter bank system.

For reconstruction of the original signal from the scaling function and wavelet coefficients, consider $f_{j+1}(t) \in \mathbf{V}_{j+1}$:

$$f_{j+1}(t) = \sum_k c_{j+1}[k] 2^{-(j+1)/2} \phi(2^{-j-1}t - k). \tag{2.53}$$

Rewriting the equation in terms of next lower scale components, we have

$$f_{j+1}(t) = 2^{-j/2} \sum_k c_j[k] \phi(2^{-j}t - k) + 2^{-j/2} \sum_k d_j[k] \psi(2^{-j}t - k). \tag{2.54}$$

Substituting eqns. (2.19) and (2.32) into (2.54), we have

$$f_{j+1}(t) = \sum_k c_j[k] \sum_n h[n] \phi(2^{-j+1}t - 2k - n) + \sum_k d_j[k] \sum_n g[n] \psi(2^{-j+1}t - 2k - n). \tag{2.55}$$

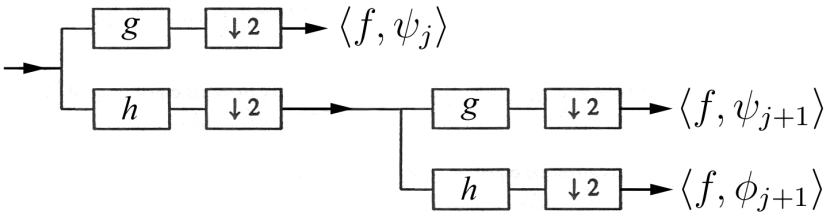
Taking the inner product $\langle f_{j+1}, \phi_{j+1,k} \rangle$ on both sides of eqn. (2.55), it simplifies to

$$c_{j+1}[k] = \sum_m h[k-2m] c_j[m] + \sum_m g[k-2m] d_j[m], \tag{2.56}$$

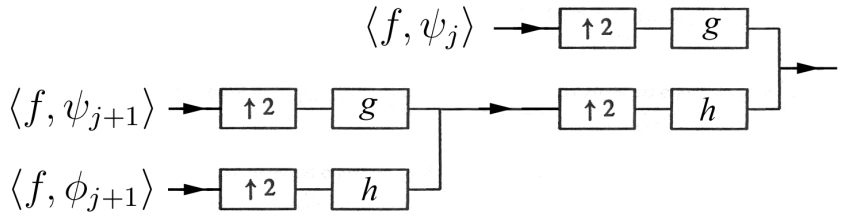
which is a reconstruction equation that constitutes upsampling the scaling function and wavelet coefficients followed by a convolution with their respective filters. Again, it is obvious that we can implement these equations through a filter bank system. The set of decomposition and reconstruction eqns. (2.51), (2.52) and (2.56) is known as Mallat's filter bank algorithm [30][66][73] for fast discrete wavelet transform, or Mallat's algorithm. Figure 2.2 illustrates the filter bank implementations of DWT and inverse DWT.

2.3 Wavelet Properties and Considerations

The wavelets are used to represent the details lost in a signal f when it is being projected onto coarser scales. As such, especially in data compression, it is desirable that



(a) Discrete dyadic wavelet transform



(b) Inverse discrete dyadic wavelet transform

Figure 2.2: Fast discrete wavelet transform using filter bank implementation.

these wavelets can efficiently approximate certain classes of functions with few non-zero coefficients as possible. Thus the design and choice of ψ must be optimized to produce a maximum number of wavelet coefficients $\langle f, \psi_{j,n} \rangle$ that are close to zero. This depends on the *regularity* of f, the *vanishing moments* of ψ and the size of its *support*.

2.3.1 Vanishing moments of ψ

A wavelet is said to have p vanishing moments if and only if its scaling function can generate polynomials of degrees smaller than or equal to p. For a wavelet with p vanishing moments,

$$\int t^k \psi(t) dt = 0, \quad 0 \leq k < p. \tag{2.57}$$

A wavelet with p vanishing moments is thus orthogonal to polynomials of degree up to $p - 1$. While this property is used to describe the approximating power of scaling functions, in the case of wavelet function, it allows the possibility to characterize the order of singularities. The number of vanishing moments is entirely determined by the coefficients of the filter h. If the Fourier transform of the wavelet is p continuously differentiable, then the following three statements are equivalent:

- the wavelet ψ has p vanishing moments;
- $\hat{\psi}(\omega)$ and its first $p-1$ derivatives are zero at $\omega = 0$;
- $\hat{h}(\omega)$ and its first $p-1$ derivatives are zero at $\omega = \pi$.

2.3.2 Compact Support of ϕ and ψ

In analysis, it is desirable that the support size of ψ be minimal to reduce number of large coefficients. If f has an isolated singularity at t_0 and if t_0 is in the support of $\psi_{j,k}$, then $|\langle f, \psi_{j,k}\rangle|$ may be large. If ψ has compact support of size K, then at each scale j there will be K wavelets $\psi_{j,k}$ whose support includes t_0. The support size of filter h is related to the support of ϕ and ψ. The scaling function is compactly supported if and only if the filter h has a finite support, and their supports are the same. If the support of the scaling function is $[N_1, N_2]$, then the wavelet support is $[\frac{N_1-N_2+1}{2}, \frac{(N_2-N_1+1)}{2}]$ of size $N_2 - N_1$. Thus to minimize the support size, the filters must be synthesized with as few non-zero coefficients as possible.

2.3.3 Regularity of ψ

The wavelet regularity is much less important than its vanishing moments. It is mostly a cosmetic influence on the error introduced by thresholding or quantizing the wavelet coefficients. Suppose error $e_{j,n}$ is introduced to each wavelet coefficients $\langle f, \psi_{j,n}\rangle$, then the reconstructed signal is given by:

$$\tilde{f} = \sum_j \sum_n (\langle f, \psi_{j,n}\rangle + e_{j,n})\psi_{j,n} = f + \sum_j \sum_n e_{j,n}\psi_{j,n}. \tag{2.58}$$

If ψ is smooth, then the reconstruction error is a smooth error. This can be important for image synthesis since a smooth error is often less visible than irregular ones, even they may have the same energy. For example, better image quality can be obtained with wavelets that are continuously differentiable than with the discontinuous Haar wavelet. Symmetric wavelets are also generally preferred so that the distortion error perceived is not skewed. It is often a misconception that increasing the number of vanishing moments in a wavelet system will increase the regularity of the wavelet and scaling functions. The following theorem shows the relation between regularity and vanishing

moments.

Theorem 2.3.1 (Tchamitchian). *Let $\hat{h}(\omega)$ be a conjugate mirror filter with p zeroes at π and satisfies the sufficient conditions in Theorem (2.1.2) . The factorization on $\hat{h}(\omega)$ gives*

$$\hat{h}(\omega) = \sqrt{2}\left(\frac{1+e^{j\omega}}{2}\right)^p \hat{q}(\omega). \tag{2.59}$$

If $\sup_{\omega\in\mathbb{R}} |\hat{q}(\omega)| = B$ then ϕ and ψ are uniformly Lipschitz α for

$$\alpha < \alpha_0 = p - \log_2 B - 1. \tag{2.60}$$

In this theorem we can see that if $\alpha_0 > 0$, then ϕ and ψ are uniformly continuous. However, increasing p does not necessarily improve the regularity of ϕ and ψ since B might increase as well. Nevertheless, for conjugate mirror filter families, B increases slower than p, thus in effect, the wavelet regularity increases with vanishing moments. Although the number of vanishing moments and the regularity of the wavelets are related, it is the vanishing moments that affect the amplitude of the wavelet coefficients at fine scales.

2.4 Summary and Remarks

We have seen in this chapter, the necessary conditions are given for functions to be qualified as wavelet for analysis purposes. We also gave an overview on different considerations of wavelet properties that would be useful for our coding applications. However, there remains an important question: *How does one construct an orthogonal wavelet basis?* We know from the discussion that if we can find a scaling function ϕ associated with a multiresolution analysis, we can find the associated orthogonal wavelets through the wavelet eqn. (2.32). So the starting point is to find a scaling function ϕ. There are essentially two approaches to finding the scaling function. The first starts from a multiresolution analysis. Usually, there exists a known, nonorthogonal basis for $\mathbf{V}_0$, and all we need to do is to use some orthogonalization procedure such as Gram-Schmidt to obtain the scaling function that generates the orthogonal basis. The second approach is to find the coefficients in the dilation eqn. (2.19) that obey appropriate

conditions, and the scaling function can then be constructed from the dilation equation. Details about the construction of wavelet functions and filters are presented in chapter 4.

3 Wavelets and Image Embedded Coding

The more sand that has escaped from the hourglass of our life, the clearer we should see through it. —Niccolo Machiavelli (1469-1527).

For rapid image transmission and reconstruction at the receiver, it is necessary to provide a coarse approximation of the original signal that progressively enhances as more bits are received, decoded and displayed. *Embedded* or *progressive* coding will allow such a feature. Bits in an embedded code are in order of importance, so that all the lower rate codes are embedded at the beginning of the data stream. For that reason, an encoder can terminate the encoding at any arbitrary point for a targeted rate or desired distortion measure. In the same way, the decoder can cease to decode the embedded bit stream at any point, reconstructing image that corresponds to all images of lower rate codes (see figure 3.1). It should be noted that for a given rate or distortion measure, a non-embedded code is usually more efficient than an embedded one since the former is free from constraints imposed by embedding properties.

In this chapter, we look at how the wavelet coefficient map provides a natural framework for generating embedded code stream. The main idea is to take advantage of prior knowledge of the locations of significant wavelets coefficients in the coarser scale

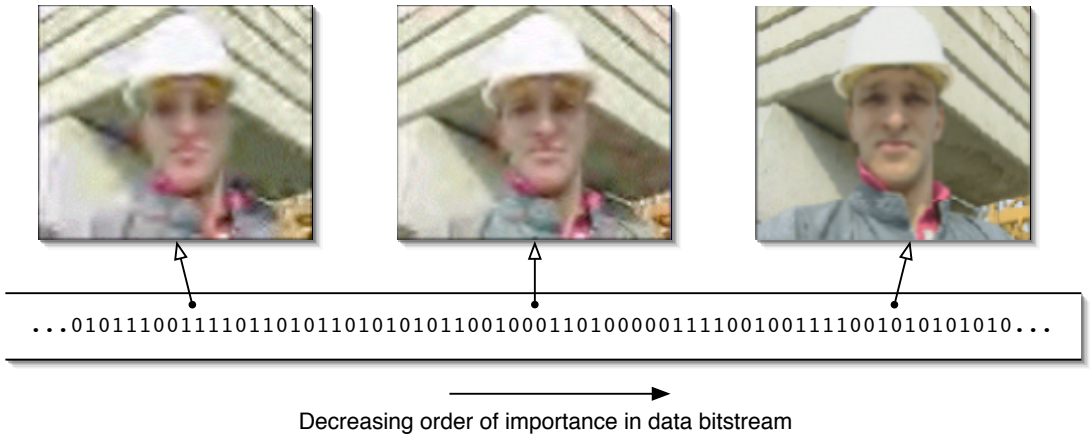


Figure 3.1: Progressive image decoding from an embedded data stream.

and efficiently predict other significant finer scale coefficients via a tree structure. In this chapter, we look at several existing embedded coding schemes for color images and present our embedded color coding technique.

3.1 Embedded Zerotree Wavelets (EZW)

Shapiro introduced the *Embedded Zerotree Wavelets* (EZW) algorithm [33] that is based on wavelets to compress 2D images and generates an embedded code stream in order of importance. The EZW produced very competitive coding performance against the main contemporary compression algorithms at that time. The main features of EZW include compact multiresolution representation of images by DWT, zero-tree coding of the significant wavelet coefficients, successive approximation and quantization of the wavelet coefficients, adaptive multi-level arithmetic coding, and capability to produce the exact targeted bit rate. Although this algorithm may not yield optimal rate-distortion, it does provide a practical and efficient coding scheme for a variety of image classes.

The core of the EZW compression lies in the exploitation of the self-similarity and decay property of wavelet coefficients across scales. This exploitation is eventually utilized in many wavelet-based image coding schemes. The EZW is designed to predict and efficiently represent groups of insignificant coefficients based on this important hypothesis:

> For a given threshold T, if a wavelet coefficient at scale j is insignificant with respect to T, i.e., $|\langle f, \psi_{j,k} \rangle| < T$, then there is a strong likelihood that finer scale wavelet

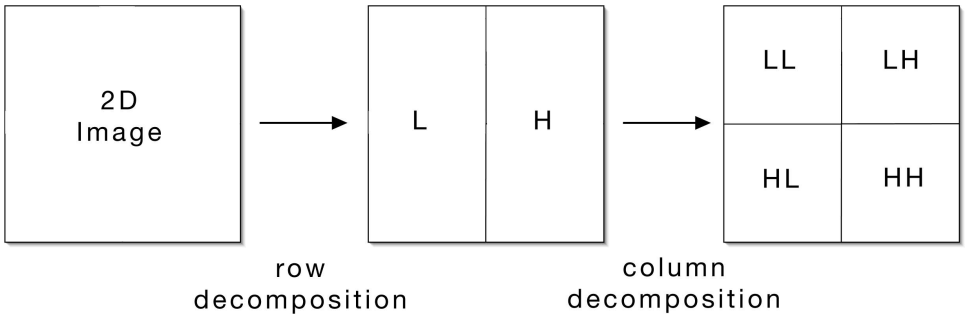
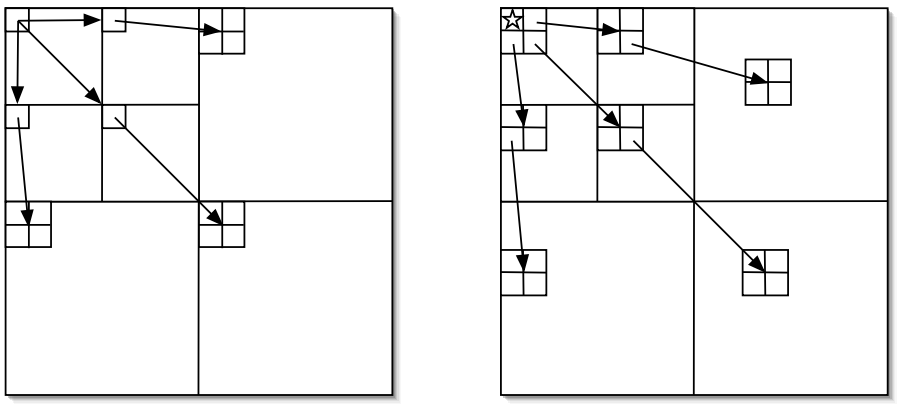


Figure 3.2: 2D wavelet transform and the resulting subband map (LL: low-low, LH: low-high, HL:high-low and HH: high-high).



(a) EZW Zerotree (b) SPIHT SOT

Figure 3.3: Spatial Orientation Trees.

coefficients in its spatial neighborhood δ will be insignificant too, i.e., $|\langle f, \psi_{j-1,k+\delta}\rangle| < T$.

This simple idea works surprisingly well for image compression since natural images generally have a predominantly low-pass spectrum, i.e., the energy of the wavelet coefficients decreases rapidly with scale. Furthermore, there is an observable relationship between clusters of insignificant wavelet coefficients across scales.

3.1.1 Zero-tree of Wavelet Coefficients

To exploit the dependency between the wavelet coefficients across different scales, Shapiro introduced an important and intuitive concept of zerotree structures (see figure 3.3(a)) which provides a natural framework for predicting insignificant coefficients across the different scales. A zerotree is a quad-tree in which all nodes have coefficients that

are equal or smaller than the root node's. There are two types of coefficient relation as in the zerotree:

- Each coefficient in the LL[1] subband has three children, each in the LH, HL and HH subbands of the same scale.
- Each coefficient, in a coarser but not the LL subband, has four children in the next finer subband of the same orientation.

The tree structure provides a compact multi-resolution representation of significance maps, which are binary maps indicating the positions of the significant coefficients. The significance $\mathcal{S}$ of a wavelet coefficient c_{ij} is a binary measure with respect to some threshold:

$$\mathcal{S}(c_{ij}, T) = \begin{cases} 0 & \text{if } |c_{ij}| < T, \\ 1 & \text{otherwise.} \end{cases} \tag{3.1}$$

A zerotree can be coded efficiently with a single symbol at the root to denote that all offsprings from the root node are zeros. EZW has assumed that there will be a very high occurrence of these zerotrees at each threshold level. When the case is true, the significance map can be compactly coded with zerotree symbols.

3.1.2 Progressive Encoding and Decoding

Like many conventional coding methods, EZW assumes that large wavelet coefficients are more important than smaller wavelet coefficients. Thus the wavelet coefficients are coded in decreasing order of magnitude in several passes. Each pass will compare the coefficient with a certain threshold and further improve the precision of the coded coefficients. It is chosen in EZW that the thresholding level to be powers of 2. This effectively allows implicit bit-wise uniform scalar quantization inside the encoding scheme. The coefficient map is scanned in a predefined zig-zag order, from high to low scale, implicitly coding many coefficient positions through the use of zerotree symbols. Several scan orders are possible as long as the lower subbands are completely scanned before going on to the higher subbands. In [33] a raster scan order is used, while in [74] some

[1]low-low subband, in 2D decomposition, four subbands are obtained at each level. The other subbands are LH (low-high), HL (high-low) and HH (high-high). See Figure 3.2 for illustrative example.

other scan orders are mentioned. The scan order seems to be of some influence in the final compression result.

It is interesting to note that the EZW algorithm reorders the wavelet coefficients through a tree structure in such an ingenious manner that they can be implicitly quantized and efficiently coded. In the next section, we see another algorithm with a more refined ordering strategy that outperforms EZW.

3.2 Set Partitioning in Hierarchical Trees (SPIHT)

In practice, it is observed in EZW that a larger portion of the bit budget is needed to code the significance map. Also, the predefined zig-zag scanning order requires that the entire lower subband to be scanned before any coefficients in the finer subband, thus delaying coding of any significant coefficients in the finer scales. The *Set Partitioning in Hierarchical Trees* (SPIHT) algorithm[2] [34] offers better rate-distortion performance than EZW. It overcomes the inefficiencies of EZW by set partitioning and reordering sets of nodes in the so-called *spatial orientation trees* or SOTs (see figure 3.3(b)). The SOT is slightly different from the tree structures used by EZW. There are two types of trees nodes used by SPIHT for the purpose of testing for tree significance:

- *Type D*, check all descendants for significance.
- *Type L*, check all of the descendants with the exception of the immediate children.

As mentioned previously, SPIHT is able to achieve a more efficient significance map coding because of the set partitioning rule, which is a sophisticated method of stepping through and partitioning sets of coefficients on the SOTs. The *set-partitioning* rule is defined to exploit the ordering in the hierarchy structure of the SOTs, with the objective of creating new partitions with large numbers of elements for insignificant subsets and only one element for a significant subset. Details of the partitioning rule is described in algorithm 3.1.

Since the ordering of DWT nodes is important for significance testing in the SPIHT, three ordered lists are maintained; namely, *list of insignificant pixels* (LIP), *list of significant pixels* (LSP) and *list of insignificant sets* (LIS). LIP contains nodes whose wavelet

[2]Software available at http://www.cipr.rpi.edu/research/SPIHT/spiht3.html

coefficients are not yet significant with respect to some threshold. Similarly, LIS contains parent nodes whose descending nodes are not yet significant. This is analogous to zerotrees in EZW. LSP contains nodes that have been found to be significant. Like EZW, SPIHT has a coding cycle for each threshold level that consist of a *sorting pass* and *refinement pass*. The sorting pass checks the significance of the LIP and LIS elements, and moves significant coefficients to the LSP. In the refinement pass, the encoder increases the bitwise precision of coefficients in the LSP by coding the next bit.

3.3 Embedded Color Image Coding

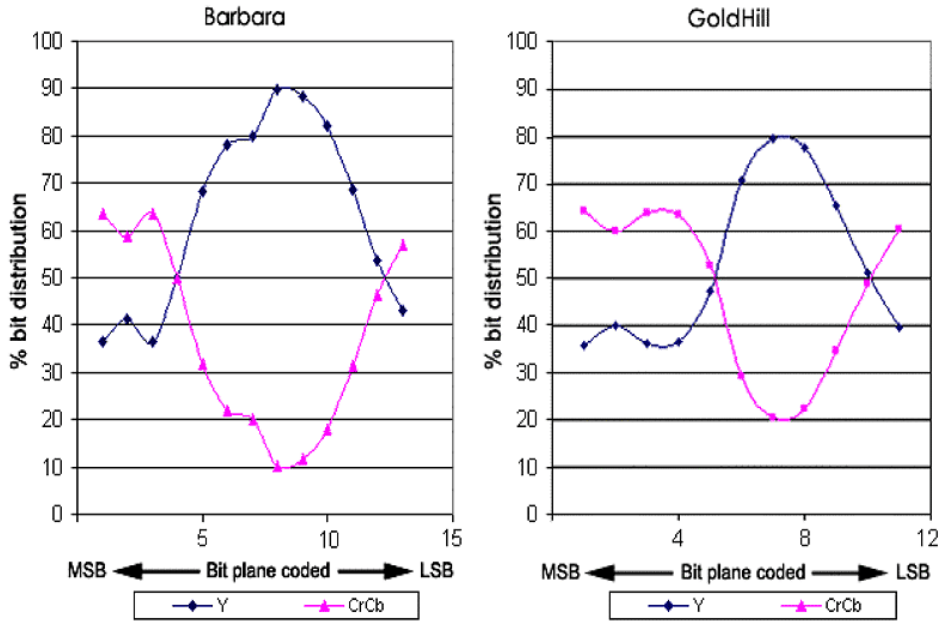
The schemes described in sections 3.1 and 3.2 are designed to code monochrome images. They are not immediately suitable for coding color images. In this section, we look at various proposals for embedded color image coding. We will also proposed our solution for color image coding using the SPIHT algorithm as a basis.

3.3.1 Representation of Color Images

According to the colorimetry theory [75][76] [46], it has been shown that any color can be represented by a combination — linear, subtractive, or additive — of three chromatic stimuli. The CIE defined the system of chromatic coordinates based on primary colors — RGB. The human visual system has much less dynamic range (acuity) for spatial variation in colour than for brightness (luminance). In other words, we are acutely aware of changes in the brightness of details than of small changes in hue. So rather than using RGB space, the system of chromatic coordinates commonly used are those that decorrelate the luminance information from the other complementary chrominance data.

In practice, images are sampled and displayed in RGB format. However for image processing or encoding purposes, the RGB components are usually transformed[3] into the statistically lesser-correlated luminance-chrominance space such as YUV and YC_bC_r. In fact, it is more efficient to encode luminance in one channel and colour information in two other channels. The two colour channels can be encoded with less bandwidth by reducing the precision representation, or the spatial resolution of the chroma data.

[3] color conversion matrices, like RGB↔YUV are usually not reversible.



(a) Barbara (b) Goldhill

Figure 3.4: Bit distribution for coding using direct color coding with 9/7 biorthogonal wavelet at six and four-level decomposition respectively.

3.3.2 Direct Color Coding with SPIHT

In direct coding of color images, we can simply use the original encoding scheme on each of the color planes, and then multiplex the three output bit streams. In this manner we are coding each plane independently. Obviously, this technique is far from being optimal since there is no exploitation of any correlation that might exist between each plane. Figure 3.4 provides information about the distribution of coded bits among luminance and chrominance nodes at each iteration using the direct method for SPIHT color coding. In the first few iterations of the SPIHT coding cycle, a greater proportion of the coded bits are allocated for chrominance nodes since they form a majority $\frac{2}{3}$ of the nodes initialized in the LIS and LIP. It is observed that the magnitude of DWT coefficients in the luminance plane is generally larger than those in chrominance planes. Thus, at the initial stages, encoded significant nodes are mostly from the luminance plane. If w_y, w_{cb} and w_{cr} are the absolute maximum of DWT coefficients in the Y, C_b , and C_r planes, respectively, and

$$M = \lfloor \log_2(w_y) \rfloor, \tag{3.2}$$

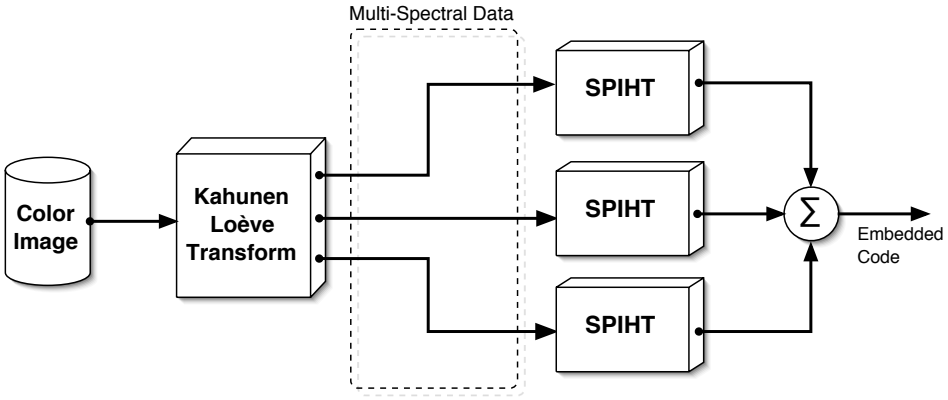


Figure 3.5: To code a color image using SPIHT, the color image is first spectrally decorrelated using KLT. The resulting three channels of spectral data is embedded coded by SPIHT independently. The bit streams are multiplexed at the output.

$$N = \lfloor \log_2 max(w_{cb}, w_{cr}) \rfloor, \tag{3.3}$$

then $M > N$ typically. The chrominance nodes will accumulate in the LIP for at least $M - N$ sorting and refinement cycles before being coded as significant and transferred to the LSP. Hence, processing the chrominance nodes in the initial stages would be a waste of bit resources, as no useful information is being conveyed to the decoder. Similarly, the argument applies to chrominance nodes in LIS.

3.3.3 Karhunen-Loève Transform and SPIHT (SPIHT+KLT)

The direct coding method is inefficient since the correlation between different color planes are not exploited for coding. In [77], Said *et al.* proposed to use the *Karhunen-Loève transform* (KLT) [78] to decorrelate[4] the color information and then encode each spectral plane independently with SPIHT. In other words, we are performing a optimal color space conversion using KLT. The required conversion is given by the eigenvectors of the covariance matrix:

$$\begin{pmatrix} \mathbf{a}\mathbf{a}^T - \bar{\mathbf{a}}^2 & \mathbf{a}\mathbf{b}^T - \bar{\mathbf{a}}\bar{\mathbf{b}} & \mathbf{a}\mathbf{c}^T - \bar{\mathbf{a}}\bar{\mathbf{c}} \\ \mathbf{b}\mathbf{a}^T - \bar{\mathbf{b}}\bar{\mathbf{a}} & \mathbf{b}\mathbf{b}^T - \bar{\mathbf{b}}^2 & \mathbf{b}\mathbf{c}^T - \bar{\mathbf{b}}\bar{\mathbf{c}} \\ \mathbf{c}\mathbf{a}^T - \bar{\mathbf{c}}\bar{\mathbf{a}} & \mathbf{c}\mathbf{b}^T - \bar{\mathbf{c}}\bar{\mathbf{b}} & \mathbf{c}\mathbf{c}^T - \bar{\mathbf{c}}^2 \end{pmatrix} \tag{3.4}$$

where **a**, **b** and **c** are row vectors denoting the lexicographic ordering of image elements in each of the three color planes. The color space conversion using KLT attempts to

[4]This idea applies equally to EZW for coding color images.

Table 3.1: KLT matrices for color space conversion from YC_bC_r

Image	Spectral Conversion Matrix
Lena	$\begin{pmatrix} 0.98 & -0.22 & -0.02 \\ -0.09 & -0.53 & 0.84 \\ 0.19 & 0.82 & -0.54 \end{pmatrix}$
Girls	$\begin{pmatrix} 0.99 & -0.06 & 0.09 \\ 0.11 & 0.71 & -0.69 \\ -0.02 & 0.70 & 0.71 \end{pmatrix}$
Pepper	$\begin{pmatrix} 0.89 & 0.12 & 0.43 \\ 0.41 & 0.18 & 0.90 \\ 0.19 & 0.98 & 0.11 \end{pmatrix}$
Barbara	$\begin{pmatrix} 0.99 & 0 & 0.03 \\ -0.02 & -0.58 & 0.82 \\ -0.02 & 0.82 & 0.58 \end{pmatrix}$

compact most of the spectral information into one plane. The main disadvantages of this method are the dependency on the source, and higher computation complexity. Table 3.1 lists the KLT matrices for color space conversion of various images from YC_bC_r format. Clearly, the Y components undergo a near identity transformation and the main conversion actually occurs in the C_bC_r plane. This suggests that we would expect performance difference to be mainly in the chrominance planes when compared to the direct coding method. We will also show in section 3.4 that the KLT method can have some adverse effect on the coding performance.

3.3.4 Color EZW (CEZW)

Luminance and chrominance values at the same spatial locations are related [3],[4],[5] such that, where the luminance values indicate large or small transitions, the corresponding chrominance values would also have similar transitions. After wavelet decomposition, these transitions will generally result in similar magnitude of wavelet coefficients. Thus, if a wavelet coefficient in luminance plane has a small magnitude, the corresponding coefficients in the chrominance planes would also be likely to have small magnitude. Shen *et al.* [79] attempt to exploit this relation by creating a comprehensively linked zerotrees across the three spectral planes (see figure 3.6). The result is a color version

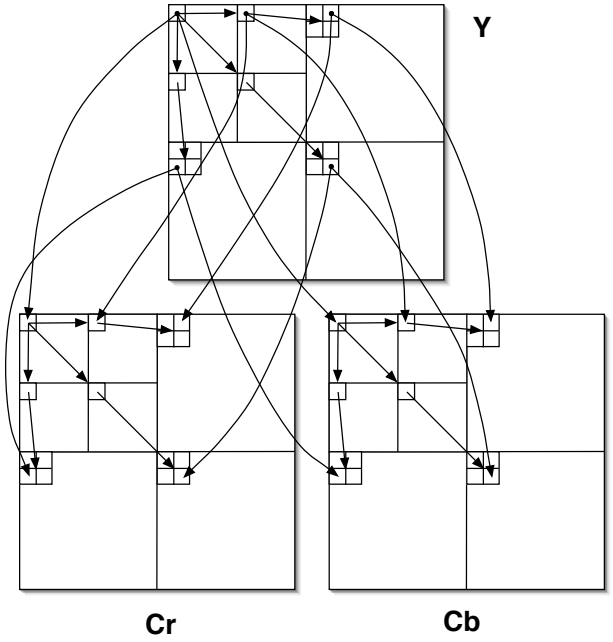
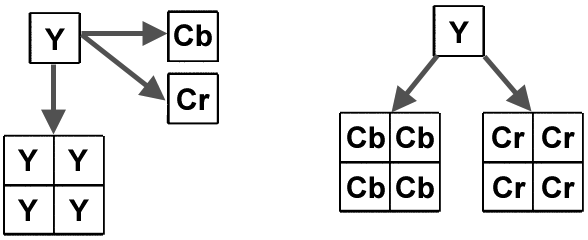


Figure 3.6: In CEZW, each luminance (Y) node has descendants in both luminance and chrominance planes – four luminance offspring in the next finer scale and two chrominance (C_b, C_r) offspring in the same scale. The chrominance nodes have only descendants in their respective planes.

of EZW — the CEZW algorithm. The CEZW scheme uses the node configuration illustrated in figure 3.7(a), where each luminance node has offspring of both luminance and chrominance types. Hence to test the significance of a tree, the encoder has to compare all the branch nodes from luminance and chrominance planes.

It is reported in [79] that the SPIHT+KLT [77] has better performance than CEZW because of its optimal color space decorrelation. However, we observed that the tree design used by CEZW is chiefly the cause of some of its inefficiency. From eqns. (3.2) and (3.3), if $(M - N)$ is large, this kind of node configuration can result in accumulation of chrominance nodes/sets in the LIP/LIS. This can lead to inefficient subsequent coding of nodes/sets in the LIP/LIS. Generally, the chrominance data are relatively smoother than those in the luminance. When a smooth function is approximated by wavelets with sufficient zeros, the expansion coefficients will have fast decay, and thus require fewer coefficients for a reasonable quality of reconstruction. From table 3.1, it is already clear that the luminance plane is generally well decorrelated from the chrominance planes



(a) (b)

Figure 3.7: Parent-children node relation in (a) CEZW and (b) CSPIHT.

for YC_bC_r color images. Hence, it is not worthwhile for the CEZW to attempt to exploit the correlation between spectral planes by expanding the SOT structure used in EZW across the spectral planes for all nodes. Instead, this comprehensive linking has resulted in much coding overheads. A further elaboration of this inefficiency will be discussed in the next section, together with our proposal of an alternate color coding scheme.

3.3.5 Color SPIHT (CSPIHT)

Here, we propose a color version of SPIHT algorithm — CSPIHT. Our objective is to increase the coding efficiency in the initial encoding iterations for color images. This would improve coding performance in low bit-rate environments whereby the encoding usually terminates after only a few iterations. Figure 3.7(b) shows the node configuration for SPIHT, in which the luminance and chrominance offspring nodes do not share any common parents. In CSPIHT, the SOT structure of SPIHT is used for each of the spectral planes. To create a single tree structure that spans across the different spectral planes, luminance nodes in the LL subband that do not have any offspring are given descendants in the LL subbands of the chrominance planes as shown in figure 3.8. In the CSPIHT scheme, only luminance nodes in the LL subband are used to initialize the LIS and LIP. The chrominance nodes will only be added to the list when they are found to be significant through their luminance parent nodes in LIS. Suppose the lowest subband has dimension X by Y; CSPIHT requires $2XY$ nodes in initialization of the LIP and LIS, whereas the SPIHT+KLT would require $3XY$ nodes for LIP (50% more than CSPIHT) and $\frac{9}{4}XY$ nodes for LIS (12.5% more than CSPIHT).

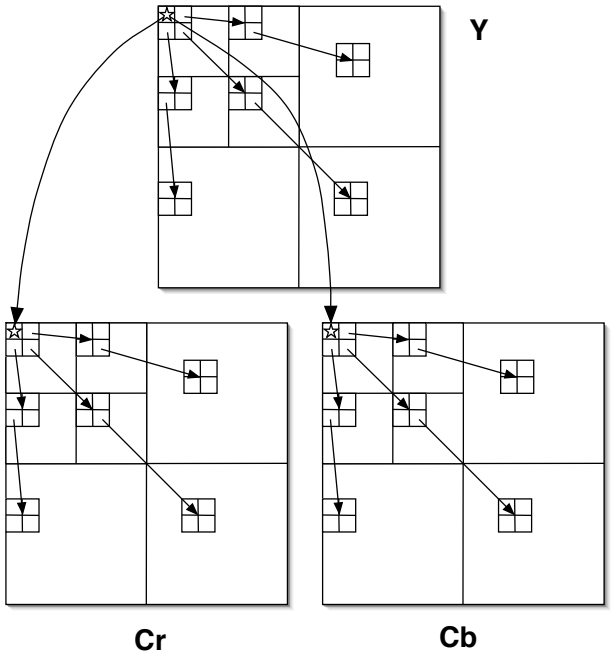
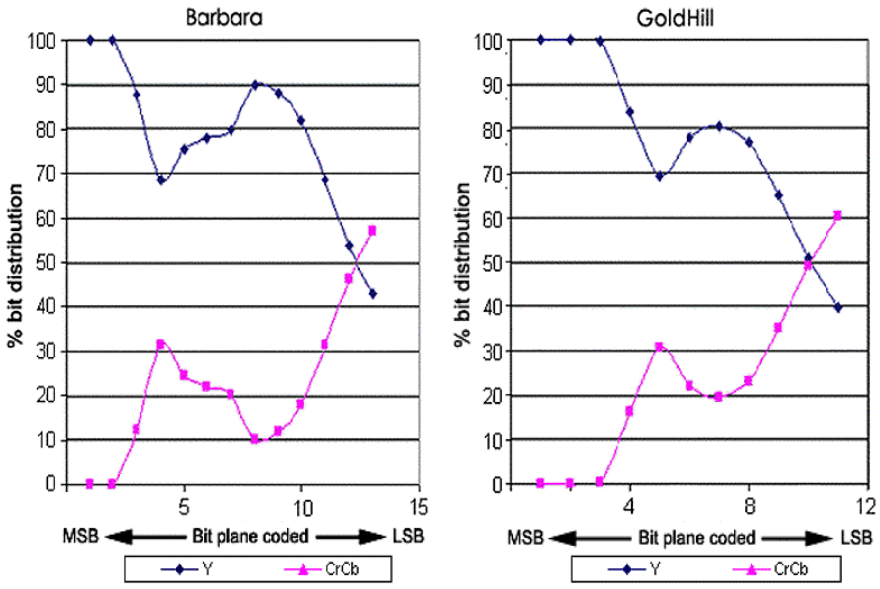


Figure 3.8: In CSPIHT, only nodes marked *star* in the lowest or LL band of the luminance (Y) plane have eight descendants in the chrominance (C_b,C_r) planes; whereas, other luminance nodes have descendants only in the luminance planes. Note that chrominance nodes marked *star* do not have any descendants.

Figure 3.9 shows the bit distribution for the CSPIHT scheme. Compared to figure 3.4, the CSPIHT scheme allocates most of the bits for luminance nodes in the initial coding phases since most of the insignificant chrominance nodes have not been included in the lists for encoding. Due to the fact that the inter-planar linkages between SOTs are confined to the LL subbands, we expect the coding advantage gained to be more significant at low bit rates and especially when the dimension of the LL subband is large.

In the previous section, we have mentioned how the node configuration used by CEZW may result in coding inefficiency. We elaborate the problem here with contrast to our proposed method. Supposing we used the CEZW tree configuration on SPIHT scheme for color coding. When a Y node has significant offsprings, its six children nodes will be evaluated and added to LIP or LSP accordingly. In the early coding stages, the Y nodes are more likely to dominate among the significant nodes because the Y nodes generally have more energy compared to their corresponding C_bC_r nodes. Most of the chrominance nodes will be added to LIP instead of LSP initially. Accumulation



(a) Barbara (b) Goldhill

Figure 3.9: Bit distribution for coding using CSPIHT with 9/7 biorthogonal wavelet six and four-level decomposition respectively.

of the C_bC_r nodes in LIP can occur if there is a large difference between the absolute maximum of the wavelet coefficients from the luminance and chrominance planes. This fast growing LIP makes the SOT of CEZW inefficient, especially when bit budget is low.

In our proposed CSPIHT scheme, only the Y nodes without offsprings are assigned C_bC_r as its children. This configuration has the advantage of allowing multiple C_r and C_b zerotree roots to be coded jointly as an insignificant set using a single symbol. This is because there exists similarity between the transformed C_r and C_b planes which share common zerotree roots and have comparable coefficient magnitudes. If a C_r node is a zerotree, the corresponding C_b node is also likely to be one too. Additionally, this node configuration delays parsing into the chrominance plane. Hence, by both delayed and joint coding of the chrominance zerotrees, CSPIHT is able avoid the accumulation of chrominance nodes in LIP and allows more resources to be spent on luminance data.

3.4 Numerical Examples

In this section, we only compare the performance of SPIHT+KLT and CSPIHT since CEZW has been shown to have poorer performance than SPIHT+KLT in [79]. We have

also explained in sections 3.3.4 and 3.3.5 that how its performance can be impaired by its SOT design. For common basis of comparison, binary-uncoded versions of SPIHT+KLT and CSPIHT are used.

PSNR values obtained by averaging MSE in the color spaces does not accurately convey the differences in the quality of the reconstructed images. We have found it more meaningful to interpret the PSNR of each color plane, which enables us to understand reconstruction error that dominates the image. PSNR readings in the Y space provide information about the reconstruction of image details while PSNR readings in C_bC_r spaces provide information about color distortion in the reconstructed image. Individual PSNR readings in RGB would not directly provide this type of information. For formality, the PSNR measurement is given as follows:

$$\mathrm{PSNR}_X = 10 \log_{10} \frac{255^2}{\mathrm{MSE}(X)} \tag{3.5}$$

where MSE$(\cdot)$ denotes mean-squared-error function and X denotes the spectral plane in measurement. Tables 3.2 and 3.3 compare the PSNR values of decoded images using SPIHT+KLT and CSPIHT schemes.

Generally for the set of 512×512 images, the difference between PSNR_Y of CSPIHT and CSPIHT+KLT is only about 0.1 dB on average. The main difference is found in the chrominance C_r and C_b planes where CSPIHT+KLT performs better. This is in agreement with the color distortion observed in CSPIHT reconstructed images as mentioned above. For 720×576 images, the CSPIHT has significantly better PSNR_Y performance than SPIHT+KLT. For the case of four- and five-level DWT decomposition, CSPIHT has significantly better PSNR performance than SPIHT+KLT, especially at low bit-rates with up to 6 dB improvement in performance. As discussed in section 3.3.5, the CSPIHT provides good overall performance for LL bands of large sizes and at lower number of DWT levels. This is clearly evident from the results in Table 3.3 and examples in figure 3.10, 3.11 and 3.12.

Interestingly, we find that the SPIHT+KLT does not usually give optimal performance. A good example is the test on *Peppers* image. In SPIHT+KLT, KLT is applied

prior to SPIHT to decorrelate the spectral components and hence minimize the overall reconstruction error when each spectral plane is coded independently. However, the application of the KLT can result in high incidences of zerotree prediction failure as shown in table 3.4. In the sorting pass, a prediction failure occurs if a type-A node in LIS is found to have significant descendants when its offspring are not significant. The coding efficiency of SPIHT depends very much on successful zerotree prediction, as more bits would be required to code a significant node where the prediction fails.

3.5 Summary and Remarks

In this chapter, we reviewed several schemes for embedded coding of color images. This includes direct coding method, color space preprocessing using KLT as proposed by Said *et al.* [77], and the CEZW by Shen *et al.* [79] which uses a comprehensively linked SOT. Based on the nature of the luminance-chrominance data and the understanding of the various problems found with the reviewed methods, we have proposed an simple color coding solution, CSPIHT, to embed both luminance and chrominance data into a single coded data stream. CSPIHT offers very comparative reconstruction quality without any extra complexity as with the SPIHT+KLT solution. The performance of the CSPIHT scheme is also shown to exceed that of the CEZW. The CSPIHT coding scheme has performance advantage in low bit-rate settings and when the wavelet LL subband has a sizeable dimension. In cases where the lowest DWT subband size is small, both CSPIHT and SPIHT+KLT have very similar performance despite the use of the KLT in the latter. Visually, the SPIHT+KLT has slightly less color distortion (via PSNR readings) than CSPIHT at lower bit rates. When the lowest subband dimension is large, CSPIHT has a better performance.

It is important to point out that the coding schemes discussed so far are based on a common assumption: *large wavelet coefficients are more important than small wavelet coefficients.* In fact, this is also a general assumption in many lossy compression applications. Although this can result in a simple and fast coding scheme, the assumption grossly ignores the complexity in human visual perception. In the following chapters, we will focus on finding a good representation and coding techniques for one of the

most perceptually important features— *edges*.

Table 3.2: Comparison between CSPIHT and SPIHT+KLT (Part 1)

BARBARA

	6-level decomposition						4-level decomposition					
	CSPIHT			SPIHT+KLT			CSPIHT			SPIHT+KLT		
bpp	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r
0.05	24.24	35.34	34.80	24.23	35.87	35.29	23.37	33.00	32.71	20.66	31.35	30.38
0.1	25.36	36.43	36.28	25.25	36.98	36.71	24.82	36.06	35.83	23.09	35.02	34.28
0.2	27.42	37.14	37.17	27.48	37.63	37.52	26.87	37.27	37.18	25.56	37.04	36.74
0.4	30.84	38.65	38.82	30.87	38.97	39.20	30.48	38.65	38.78	29.22	38.69	38.80
0.6	33.11	39.52	39.97	33.16	39.92	40.39	32.82	39.48	39.88	32.07	39.65	39.97
0.8	35.04	40.28	40.82	35.14	40.51	41.06	34.81	40.26	40.78	34.17	40.33	40.79
1.0	36.59	40.81	41.35	36.80	40.98	41.64	36.40	40.79	41.30	35.99	40.80	41.40

GIRLS

	6-level decomposition						4-level decomposition					
	CSPIHT			SPIHT+KLT			CSPIHT			SPIHT+KLT		
bpp	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r
0.05	29.00	41.17	40.42	28.96	41.78	40.02	27.79	36.66	35.19	21.09	32.61	32.43
0.1	30.66	42.53	42.24	30.65	43.59	41.74	29.73	42.29	41.28	25.20	37.35	36.13
0.2	32.80	44.38	44.17	32.75	44.87	43.46	32.17	44.28	43.74	30.03	43.44	41.52
0.4	35.07	46.27	46.22	35.08	46.36	45.20	34.67	45.93	45.78	33.59	46.10	44.95
0.6	36.75	47.36	47.44	36.80	47.38	46.18	36.46	47.38	47.52	35.70	47.12	45.83
0.8	38.29	48.21	48.24	38.35	48.01	46.84	38.04	48.05	48.12	37.28	47.93	46.76
1.0	39.62	48.85	49.06	39.57	48.72	47.44	39.42	48.80	48.99	38.77	48.30	47.10

PEPPERS

	6-level decomposition						4-level decomposition					
	CSPIHT			SPIHT+KLT			CSPIHT			SPIHT+KLT		
bpp	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r
0.05	24.90	32.73	30.81	24.83	33.31	30.84	23.30	29.72	28.39	20.31	28.63	25.55
0.1	28.13	34.18	32.90	27.91	34.96	32.01	26.98	33.91	32.01	23.30	32.75	30.41
0.2	31.95	36.36	35.99	31.23	37.55	35.39	31.30	36.11	35.40	28.91	35.67	32.48
0.4	35.43	39.84	39.86	34.90	40.08	39.65	35.17	39.64	39.49	33.70	39.50	38.76
0.6	37.64	42.47	42.49	36.91	42.16	41.89	37.40	42.28	42.26	36.14	41.50	40.64
0.8	38.93	43.66	43.78	38.44	43.26	42.92	38.75	43.54	43.65	37.82	43.00	42.62
1.0	40.06	44.80	44.76	39.34	44.40	43.62	39.88	44.63	44.61	39.02	43.96	43.40

LENA

	6-level decomposition						4-level decomposition					
	CSPIHT			SPIHT+KLT			CSPIHT			SPIHT+KLT		
bpp	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r	Y	C_b	C_r
0.05	27.77	34.11	34.28	27.83	35.71	34.47	25.41	32.12	32.16	21.80	31.74	30.52
0.1	30.14	35.64	35.68	30.28	36.81	35.92	28.71	35.08	35.09	25.07	34.56	33.59
0.2	32.63	37.26	37.18	32.93	37.86	37.61	32.05	36.88	36.91	30.35	37.13	36.40
0.4	35.59	38.75	38.61	35.76	38.91	39.18	35.28	38.58	38.41	34.60	38.58	38.55
0.6	37.20	39.66	39.72	37.39	39.75	40.14	37.05	39.55	39.58	36.57	39.51	39.80
0.8	38.36	40.11	40.21	38.51	40.24	40.77	38.18	40.05	40.15	38.03	39.96	40.47
1.0	38.86	40.67	40.86	38.92	40.91	41.30	38.81	40.58	40.77	38.76	40.56	41.05

Table 3.3: Comparison between CSPIHT and SPIHT+KLT (Part 2)

GOLDHILL
4-level decomposition

	CSPIHT			SPIHT+KLT		
bpp	Y	C_b	C_r	Y	C_b	C_r
0.05	26.48	34.13	33.41	22.58	33.45	30.74
0.1	28.52	36.83	35.31	26.57	35.45	34.26
0.2	30.63	38.41	36.76	29.54	38.20	36.60
0.4	32.99	39.66	37.91	32.39	39.42	37.83
0.6	34.62	40.37	38.63	34.07	40.01	38.51
0.8	35.78	40.87	39.10	35.38	40.64	39.04
1.0	36.75	41.22	39.44	36.36	41.03	39.40

BOAT
4-level decomposition

	CSPIHT			SPIHT+KLT		
bpp	Y	C_b	C_r	Y	C_b	C_r
0.05	25.00	33.93	32.51	21.69	30.83	29.58
0.1	27.52	37.16	35.60	25.59	35.14	33.81
0.2	30.77	39.08	37.51	29.68	38.78	37.21
0.4	34.32	40.66	39.62	33.57	40.71	39.58
0.6	36.75	41.85	41.16	36.19	41.90	40.96
0.8	38.27	42.67	42.09	37.84	42.56	41.94
1.0	39.48	43.59	43.10	39.15	43.31	42.86

TOYS
4-level decomposition

	CSPIHT			SPIHT+KLT		
bpp	Y	C_b	C_r	Y	C_b	C_r
0.05	28.18	27.54	30.42	17.72	22.90	25.30
0.1	33.87	33.52	37.90	21.86	26.00	29.56
0.2	37.33	36.00	41.17	33.01	33.67	38.68
0.3	38.54	37.31	42.09	37.21	35.60	40.52
0.4	39.28	38.16	42.72	38.35	36.92	41.95

Table 3.4: PSNR Performance and Incidence Count of *Failed Predictions* (FP)

6 level decomposition

		CSPIHT			SPIHT+KLT		
Image	bpp	Y	C_bC_r	FP	Y	C_bC_r	FP
Pepper	0.1	28.13	33.54	**276**	27.91	33.49	**319**
	0.2	31.95	36.18	**398**	31.23	36.47	**508**
	0.4	35.43	39.85	**588**	34.90	39.87	**715**

(a) *Barbara*, SPIHT+KLT

(b) *Barbara*, CSPIHT

(c) *Lena*, SPIHT+KLT

(d) *Lena*, CSPIHT

(e) *Girls*, SPIHT+KLT

(f) *Girls*, CSPIHT

Figure 3.10: Embedded color image reconstruction comparison part-I, 0.1bpp.

(a) *Pepper*, SPIHT+KLT

(b) *Pepper*, CSPIHT

(c) *Boat*, SPIHT+KLT

(d) *Boat*, CSPIHT

(e) *Goldhill*, SPIHT+KLT

(f) *Goldhill*, CSPIHT

Figure 3.11: Embedded color image reconstruction comparison part-II, 0.1bpp.

(a) *Toys*, SPIHT+KLT

(b) *Toys*, CSPIHT

Figure 3.12: Embedded color image reconstruction comparison part-III, 0.1bpp.

Algorithm 3.1: SPIHT algorithm

```
begin
    Compute the threshold T;
    Initialize LIP to all root nodes coefficients;
    Initialize LIS to all tree roots as Type D;
    Initialize LSP as an empty set;
    repeat
        forall nodes in LIP do
            if significant then
                output '1' and sign bit, and move it to LSP;
            else
                output '0';
            end
        end
        forall trees in LIS do
            if type D tree then
                if significant then
                    output '1';
                    for each children do
                        if significant then
                            output '1' and sign bit, add it to LSP;
                        else
                            output '0', add it to the end of LIP;
                        end
                        if have descendants then
                            move tree to the end of LIS as type L;
                        else
                            remove from LIS;
                        end
                    end
                else
                    output '0';
                end
            end
            if type L tree then
                if significant then
                    output '1', add each children to the end of LIS as type D;
                    remove tree from LIS;
                else
                    output 0;
                end
            end
        end
        decrease threshold;
    until T = 0;
end
```

Algorithm 3.2: CSPIHT algorithm

```
begin
    Compute the threshold T = ⌊log2 max(i,j,k)(|w_{i,j,k}|)⌋;
    Initialize LIP to all root nodes coefficients in luminance (k = 0) plane;
    Initialize LIS to all tree roots in luminance (k = 0) plane as Type D;
    Initialize LSP as an empty set;
    repeat
        forall nodes (i, j, k) in LIP do
            if significant then
                output '1' and sign bit, and move (i, j, k) to LSP;
            else
                output '0';
            end
        end
        forall trees (i, j, k) in LIS do
            if type D tree then
                if significant then
                    output '1';
                    for each children do
                        if significant then
                            output '1' and sign bit, add it to LSP;
                        else
                            output '0', add it to the end of LIP;
                        end
                        if have descendants then
                            move tree (i, j, k) to the end of LIS as type L;
                        else
                            remove tree (i, j, k) from LIS;
                        end
                    end
                else
                    output '0';
                end
            end
            if type L tree then
                if significant then
                    output '1', add each children to the end of LIS as type D;
                    remove tree (i, j, k) from LIS;
                else
                    output 0;
                end
            end
        end
        decrease threshold;
    until T = 0;
end
```

4 Analysis and Synthesis of Finite Signals

...the habit of literature is also the habit of interpolating circumstantial details and accentuating certain emphases. —Jorge Luis Borges (1899-1986).

When using the wavelet transform as a mathematical analysis tool, the infinite and continuous descriptions of $t \in \mathbb{R}$ are appropriate. In practical signal processing and numerical analysis, the signal or function $f[n]$ is discrete and most probably finite. If the discrete signal has finite support on $[0, N]$, i.e., $N+1$ samples, then ideally one should have $N + 1$ nonzero transform coefficients $\langle f, \phi_{j,k} \rangle$. However, the discrete wavelet transform will typically produce more than $N + 1$ coefficients, since we are using basis in $\mathbf{L}^2(\mathbb{R})$ to represent a signal in $\mathbf{L}^2[0, N]$. This is certainly undesirable for coding applications. Another problem arises from computing the expression $\sum_k h_k x(n - k)$, where the values of x outside its interval $[0, N]$ is required for filtering with h. Generally, there are two possible solutions. The signal can be extrapolated beyond its boundary; or special boundary filters, which do not overlap the signal borders, can be applied. With regards to data compression, for either of the methods, it is desired that there should be no data

redundancy or large coefficients created near the boundaries.

In this chapter, different methods are reviewed for performing discrete wavelet transform on finite data sets. There are essentially two approaches: signal extension and adapting filters to the interval. We review some of the more common techniques based on these approaches. We also introduce our interval filter designs that will allow robust decomposition of finite signals. Ideas introduced in this chapter are important for efficient coding of signals and images, which are discussed in subsequent chapters.

4.1 Signal Extension and Extrapolation

There exist several techniques to extend or extrapolate a finite signal beyond its bound. The common methods are: *zero-padding*, *periodic extension*, *symmetric extension* and *polynomial extrapolation*. This simplest is zero-padding which assumes the signal outside its support to be zeros but it creates artificial discontinuities at the borders. Thus, this method is not usually used in image processing. When performing wavelet transform on finite signals like images, one of these methods are employed in the filtering process. These extension and extrapolation methods could be also be viewed as wavelet transforms with appropriately re-mapped wavelet bases.

4.1.1 Periodic Extension or Cyclic Wavelet

Periodic signal extension is equivalent to periodizing the wavelets $\psi_{j,n}$. The periodic extension of $f \in \mathbf{L}^2[0, N]$ over $\mathbb{R}$ can be defined as:

$$f^{per}(t) = \sum_{k=-\infty}^{+\infty} f(t + kN), \quad \in \mathbf{L}^2(\mathbb{R}). \tag{4.1}$$

Since f^{per} is defined on the whole line, we can perform decomposition on it with $\psi \in \mathbf{L}^2(\mathbb{R})$. In a different perspective, decomposing f^{per} in a regular basis of $\mathbf{L}^2(\mathbb{R})$ is equivalent to decomposition f in a periodic wavelet basis:

$$\int_0^N f(t)\psi_{j,n}^{per}(t)dt = \int_{-\infty}^{+\infty} f^{per}(t)\psi_{j,n}(t)dt \tag{4.2}$$

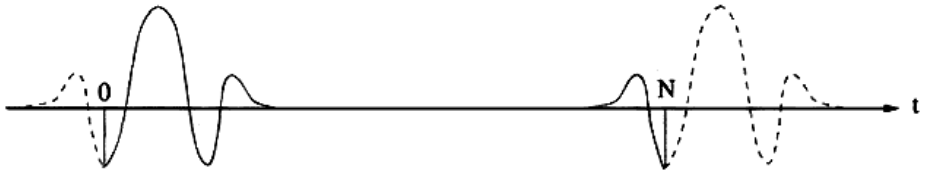


Figure 4.1: On the interval $[0, N]$, a periodic wavelet ψ^{per} has two disjoint components near $t = 0$ and $t = N$.

where the periodic wavelet is defined as:

$$\psi^{per}_{j,n}(t) = \frac{1}{\sqrt{2^j}} \sum_{k=-\infty}^{+\infty} \psi(\frac{t - 2^j n + kN}{2^j}) \tag{4.3}$$

If the support of $\psi_{j,n}$ is in $[0, N]$ then $\psi^{per}_{j,n}(t) = \psi_{j,n}(t)$ for $t \in [0, N]$. This periodization modifies only the boundary wavelets whose supports overlap the boundaries at $t = 0$ or $t = N$. These periodic wavelets together with periodized scaling functions $\phi^{per}_{j,n}$ generate an orthogonal basis of $\mathbf{L}^2[0, N]$. For periodic wavelet decomposition of $f \in \mathbf{L}^2[0, N]$, we are guaranteed periodicity of $2^{-j}N$ in the scaling and wavelet coefficients

$$c_j[n + 2^{-j}N] = c_j[n] = \langle f, \phi^{per}_{j,n} \rangle,$$

$$d_j[n + 2^{-j}N] = d_j[n] = \langle f, \psi^{per}_{j,n} \rangle.$$

The periodicity in the solutions implies that only $N+1$ coefficients is required for perfect reconstruction, i.e., coefficients outside $[0, N]$ can be discarded. Note that the periodic wavelets ψ^{per} have disjoint components (see figure 4.1) with no vanishing moments. If $f(0) \neq f(N)$, discontinuities are artificially created at the borders and large amplitude wavelet coefficients can be generated in the neighborhood of $t = 0$ and $t = N$.

4.1.2 Symmetric Extension or Folded Wavelet

To reduce the effect of artificial discontinuities created at the borders, the signal f can be extended symmetrically:

$$f^{sym}(t) = \sum_{k=-\infty}^{+\infty} f(t - 2kN) + \sum_{k=-\infty}^{+\infty} f(2kN - t) \tag{4.4}$$

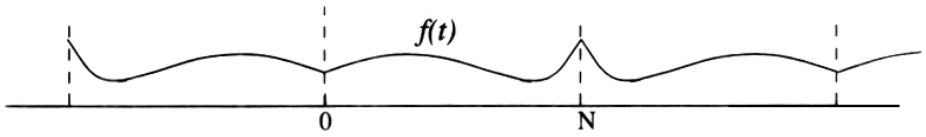


Figure 4.2: The extended signal, f^{sym} is $2N$ periodic, symmetric about $t = 0$ and $t = N$. On the interval $[0, N]$, $f^{sym} = f$.

This leads to zeroth order continuity at the borders (see figure 4.2). If f is continuously differentiable then f^{sym} is continuous at $t = 0$ and $t = N$. Hence, symmetric extension can produce smaller border wavelet coefficients than f^{per}. However, discontinuities of the first order derivatives still exist. If $f'(0) \neq 0$ and $f'(N) \neq 0$, the function derivatives at $t = 0$ and $t = N$ are discontinuous . Thus, large amplitude wavelet coefficients can still be produced near the border.

Decomposing f^{sym} in wavelet basis $\psi_{j,n} \in \mathbf{L}^2(\mathbb{R})$ is equivalent to decomposing f on folded wavelet basis $\psi^{fold}_{j,n} \in \mathbf{L}^2[0, N]$:

$$\int_0^N f(t)\psi^{fold}_{j,n}(t)dt = \int_{-\infty}^{+\infty} f^{sym}(t)\psi_{j,n}(t)dt \tag{4.5}$$

The folded wavelets $\psi^{fold}_{j,n}$ can be constructed from $\psi(t)$ that are either symmetric or antisymmetric with respect to $t = \frac{1}{2}$. The Haar wavelet is the only compactly supported and orthogonal basis that is symmetric or antisymmetric. Thus biorthogonal wavelets are usually utilized for folded wavelet transform. Note that the treatment of borders is actually more complicated, in which the symmetric and antisymmetric cases are to be considered separately. An important consequence of symmetric extension is that, if a filter is symmetric (or antisymmetric) and appropriate symmetric extension on a signal, the filtered output will be symmetric (or antisymmetric). This is sufficient to guarantee non redundant decomposition.

4.1.3 Polynomial or Wavelet Extrapolation

To obtain smoother extensions, the signal can be polynomial extrapolated at the borders. The principle is to fit a polynomial to existing data and extend that polynomial outside the bound. Thus, only discontinuities at higher order derivatives will be created.

The degree k of polynomial used for extrapolation depends on the number of vanishing moments p of the wavelets. Typically, it is chosen to be $k = p - 1$. If the signal is sufficiently regular, one can used Lagrange polynomial extrapolation. For stability, especially in the presence of noise and when using longer filters, least-square fitting polynomial extrapolation is preferred. Consider k^{th} degree polynomial approximation

$$y_i = \sum_{n=0}^{k} a_n x_i^n \tag{4.6}$$

where a_n are the polynomial coefficients. In matrix notation, the equation for a polynomial fit is given by:

$$\begin{bmatrix} 1 & x_1 & \dots & x_1^k \\ 1 & x_2 & \dots & x_2^k \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \dots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_k \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \tag{4.7}$$

i.e., $X\mathbf{a} = \mathbf{b}$. This can be solved in the least square sense to yield

$$\mathbf{a} = (X^T X)^{-1} X^T \mathbf{b} \tag{4.8}$$

The problem of polynomial extension is that in order to have perfect reconstruction, N coefficients is insufficient. As an example, we consider Daubechies wavelet with $p = 2$, and 2^{nd}-order polynomial extension. For wavelet decomposition of $\{x_n\}_{0 \le n < N}$, we obtain a $\frac{N}{2} \times (2p - 2 + N)$ block Toeplitz form:

$$\begin{bmatrix} c_0 \\ d_0 \\ c_1 \\ d_1 \\ \vdots \\ c_{\frac{N}{2}-2} \\ d_{\frac{N}{2}-2} \\ c_{\frac{N}{2}-1} \\ d_{\frac{N}{2}-1} \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & h_3 & & & & & & \\ g_0 & g_1 & g_2 & g_3 & & & & & & \\ & & \ddots & & & & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & & & \\ & & g_0 & g_1 & g_2 & g_3 & & & & \\ & & & & h_0 & h_1 & h_2 & h_3 & & \\ & & & & g_0 & g_1 & g_2 & g_3 & & \\ & & & & & & h_0 & h_1 & \mathbf{h_2} & \mathbf{h_3} \\ & & & & & & g_0 & g_1 & \mathbf{g_2} & \mathbf{g_3} \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-3} \\ x_{N-2} \\ x_{N-1} \\ \mathbf{x_N} \\ \mathbf{x_{N+1}} \end{bmatrix} \tag{4.9}$$

where x_N and x_{N+1} are extrapolated function coefficients. And the reconstruction is given by

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-3} \\ x_{N-2} \\ x_{N-1} \end{bmatrix} = \begin{bmatrix} \mathbf{h_2} & \mathbf{g_2} & h_0 & g_0 & & & & \\ \mathbf{h_3} & \mathbf{g_3} & h_1 & g_1 & & & & \\ & & h_2 & g_2 & & & & \\ & & h_3 & g_3 & & & & \\ & & & & \ddots & & & \\ & & & & h_0 & g_0 & & \\ & & & & h_1 & g_1 & & \\ & & & & h_2 & g_2 & h_0 & g_0 \\ & & & & h_3 & g_3 & h_1 & g_1 \end{bmatrix} \begin{bmatrix} c_{-2} \\ d_{-2} \\ c_{-1} \\ d_{-1} \\ c_0 \\ d_0 \\ \vdots \\ c_{\frac{N}{2}-1} \\ d_{\frac{N}{2}-1} \end{bmatrix} \tag{4.10}$$

In order to reconstruct the original coefficients, we find that we require low-pass and high-pass coefficients outside the support $[0, N/2)$ since x_0 and x_1 have dependency on c_{-2}, d_{-2}, c_{-1} and d_{-1} . Unlike periodic and symmetric extensions, where the transformed coefficients are also periodic or symmetric, there is no clear indication of how we could obtain these out-of-bound coefficient values through some inherent properties of the transformed coefficients if we want to have a non-expansive decomposition and yet, perfect reconstruction. Nevertheless, for orthogonal wavelets, it is still possible to recover the original function using only N coefficients using the double-shift orthonormality property of wavelet filters in the time domain. Details of computing the extrapolation for orthogonal wavelet filters can be found in [80] where the polynomial extension method is exploited in ENO wavelets (see section 5.4.3).

Williams *et al.* [81] proposed a similar solution to extrapolate the signal using wavelets. The idea is to first project the function onto the space spanned by $\phi_{0,k}$ and then perform the polynomial extrapolation. Thus replacing X in eqn. (4.8) with

$$X = \begin{bmatrix} \mu_0^0 & \mu_0^1 & \cdots & \mu_0^{p-1} \\ \mu_1^0 & \mu_1^1 & \cdots & \mu_1^{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{2p-1}^0 & \mu_{2p-1}^1 & \cdots & \mu_{2p-1}^{p-1} \end{bmatrix} \tag{4.11}$$

where are $\{\mu_k^l\}_{0 \le l,k<p}$ are the moments of the scaling function given by the following set of equations:

$$\mu_0^0 = \int_{-\infty}^{+\infty} \phi(x)dx = 1, \tag{4.12}$$

Table 4.1: Bounded wavelet transformation matrices using wavelet extrapolation

Vanishing moments	Wavelet extrapolation matrices for Daubiechies Wavelets
$p = 2$	(low-pass) $\begin{bmatrix} 0.4380 & 0.8365 & 0.224 & -0.1294 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.4380 & 0.8365 & 0.224 & -0.1294 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.4380 & 0.8365 & 0.224 & -0.1294 \\ 0 & 0 & 0 & 0 & \mathbf{-0.0085} & \mathbf{0.0129} & \mathbf{0.5174} & \mathbf{0.8924} \end{bmatrix}$ (high-pass) $\begin{bmatrix} \mathbf{0.4441} & \mathbf{-0.6727} & \mathbf{0.0129} & \mathbf{0.2156} & 0 & 0 & 0 & 0 \\ -0.1294 & -0.224 & 0.8365 & -0.4380 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1294 & -0.224 & 0.8365 & -0.4380 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1294 & -0.224 & 0.8365 & -0.4380 \end{bmatrix}$
$p = 4$	(low-pass) $\begin{bmatrix} -0.0106 & 0.0329 & 0.0308 & -0.1870 & -0.0280 & 0.6309 & 0.7148 & 0.2304 \\ \mathbf{-0.0023} & \mathbf{0.0039} & \mathbf{0.2330} & \mathbf{0.7128} & \mathbf{0.6249} & \mathbf{-0.0332} & \mathbf{-0.1826} & \mathbf{0.0578} \\ \mathbf{0.0135} & \mathbf{-0.0228} & \mathbf{-0.0153} & \mathbf{0.0121} & \mathbf{0.2657} & \mathbf{0.7452} & \mathbf{0.6041} & \mathbf{-0.1882} \\ \mathbf{-0.0464} & \mathbf{0.0784} & \mathbf{0.0522} & \mathbf{-0.0420} & \mathbf{-0.1216} & \mathbf{-0.1035} & \mathbf{0.3254} & \mathbf{1.2717} \end{bmatrix}$ (high-pass) $\begin{bmatrix} \mathbf{0.0306} & \mathbf{-0.0337} & \mathbf{-0.2653} & \mathbf{0.7128} & \mathbf{-0.5946} & \mathbf{0.0232} & \mathbf{0.2008} & \mathbf{-0.0738} \\ \mathbf{0.0019} & \mathbf{0.0155} & \mathbf{-0.0004} & \mathbf{-0.0255} & \mathbf{-0.2696} & \mathbf{0.6939} & \mathbf{-0.5812} & \mathbf{0.1654} \\ \mathbf{0.0643} & \mathbf{-0.0909} & \mathbf{-0.0698} & \mathbf{0.0345} & \mathbf{0.1285} & \mathbf{0.1191} & \mathbf{-0.3174} & \mathbf{0.1317} \\ -0.2304 & 0.7148 & -0.6309 & -0.0280 & 0.1870 & 0.0308 & -0.0329 & -0.0106 \end{bmatrix}$

$$\mu_0^l = \frac{1}{2(2^l - 1)} \sum_{i=0}^{l-1} \binom{l}{i} \left(\sum_{k=0}^{2p-1} c[k] k^{l-i} \right) \mu_0^i, \tag{4.13}$$

$$\mu_k^l = \sum_{i=0}^{l} \binom{l}{i} k^{l-i} \mu_0^i. \tag{4.14}$$

Thus the scaling functions at each scale are extrapolated, considering the left boundary, as

$$c_j[k] = \sum_{i=0}^{2p-1} v_{k+2p-2,i} c_j[i], \quad k = -2p+2, -2p+3, \ldots, -1 \tag{4.15}$$

where $v_{k,i}$ are entries to the $2p \times 2p$ matrix $A^T(A^TA)^{-1}A^T$.

An advantage of the wavelet extrapolation method is that we can have a closed form of the dyadic decomposition and reconstruction equation:

$$c_{j-1}[n] = \sum_{k=0}^{2n+2p-1} h[k-2n] c_j[k] + \sum_{k=0}^{2p-1} \Theta_{2n,k} c_j[k] \tag{4.16}$$

Table 4.2: Condition number for wavelet transform matrices

Vanishing Moment p	Condition Number for Extrapolated Wavelets	Condition Number for Interval Wavelets
2	1.8594	1.0000
3	3.6392	1.0000
4	12.8647	1.0000
5	71.9361	1.0000
6	560.9095	1.0000

$$d_{j-1}[n] = \sum_{k=0}^{2n+2p-1} g[k-2n]c_j[k] + \sum_{k=0}^{2p-1} \Delta_{2n,k}c_j[k] \tag{4.17}$$

where $n = -\frac{2p}{2}+1, -\frac{2p}{2}+2, \ldots, -1$ and

$$\Theta_{m,n} = \sum_{k=m+2p-2}^{2p-1} v_{k,n}h[k-m], \quad \Delta_{m,n} = \sum_{k=m+2p-2}^{2p-1} v_{k,n}g[k-m]$$

In wavelet extrapolation, one can simply pre-calculate the filters at the boundaries and perform the usual recursive dyadic transform on signals. In fact, wavelet extrapolation method can be seen as constructing boundary-adapted wavelets, since eqns. (4.16) and (4.17) can be written in the form of bounded Toeplitz matrices. Table 4.1 lists the transform matrices for the case of Daubechies wavelets. However, for inverse transform, it is not as straight forward as simply applying the adjoint of the forward transform matrix. The out-of-bound coefficients have to be recovered before applying the inverse transform. Also, the decomposition matrices constructed by wavelet extrapolation could have stability issues. Table 4.2 lists the condition numbers on extrapolated wavelet transform matrices. A condition number is basically a measure of stability or sensitivity of a matrix to numerical operations. In other words, we may not be able to trust the results of computations on an ill-conditioned matrix. Matrices with condition numbers near 1 are said to be *well-conditioned*. In the next section, we will look at a better solution to construct the interval wavelets which give a more stable solution.

4.2 Wavelets on the Interval [0,N]

Both periodic wavelets ψ^{per} and folded wavelets ψ^{fold} have zero and one vanishing moment, respectively. These boundary wavelets are unable to exploit the signal regularity near the borders. Polynomial extensions can produce smoother extension across the borders. However, as mentioned in section 4.1.3, direct extrapolation will result in data redundancy in wavelet transform. In a different approach special boundary wavelets ψ^{left} and ψ^{right} are constructed to replace wavelets ψ that overlap the borders, such that the transformation matrix is bounded on the signal interval. It is possible to design these boundary wavelets such that they could have as many vanishing moments as the *interior* wavelets ψ.

There exist quite a few methods for constructing wavelet filters for bounded intervals such as Herley's construction [82], Chebyshev [83] and Legendre [84] polynomials. Chebyshev-polynomial wavelets require weights in their scalar products, which can lead to difficulties in interpretation of the relative significance of their coefficients. The Legendre-polynomial wavelets only decay roughly as x^{-1}. The first construction for orthonormal wavelet bases on $[0, 1]$ was first introduced by Meyer [85] and subsequently refined by Cohen *et al.* [86]. The construction was later formalized by Chyzak *et al.* [87].

For clarity of discussion in this section, we will make clear some of the terminology mentioned here. The terms '*boundary*' and '*edge*' are used interchangeably as descriptors to denote the appropriate entities which have support overlapping the signal borders, e.g. boundary filters, boundary wavelets, edge functions, boundary functions. The term '*interior*' denotes functions, filters or wavelets whose support do not overlap the borders. These would be the standard scaling ϕ and wavelet ψ functions that are defined on the whole line. '*Interval functions*' ψ^{int} is a collective term for both the boundary and interior functions. Thus, we have interval scaling functions

$$\phi^{\text{int}} = \{\phi^{\text{left}}, \phi, \phi^{\text{right}}\} \tag{4.18}$$

and interval wavelets

$$\psi^{\text{int}} = \{\psi^{\text{left}}, \psi, \psi^{\text{right}}\}. \tag{4.19}$$

However, for simplification, the term 'interval wavelets' is often used for denoting both scaling and wavelet functions on the interval.

4.2.1 Boundary Wavelets with Vanishing Moments

As for the whole line, we desire to have the interval scaling functions to generate any polynomials, up to a certain degree, on the interval. Two sets of p border scaling functions, $\{\phi_n^{left}\}_{0\leq n<p}$ and $\{\phi_n^{right}\}_{0\leq n<p}$, are constructed for each respective border such that any k^{th} degree polynomials $\{\theta_k\}_{0\leq k<p}$ on the interval can be written as a combination of the corresponding interior and border scaling functions :

$$\theta_k(t)\mathbf{1}_{[0,N]}(t) = \sum_{n=0}^{p-1} a_n \phi_n^{left}(t) + \sum_{n=p}^{N-p-1} n^k \phi(t-n) + \sum_{n=0}^{p-1} b_n \phi_n^{right}(t) \tag{4.20}$$

The boundary wavelets computed from these interval scaling functions will have the same number of vanishing moments as the original ψ. The result arises from the following theorem [88][89]:

Theorem 4.2.1 (Fix, Strang)**.** *Suppose that ψ has an exponential decay. This wavelet has p vanishing moments if and only if for any $0 \leq k < p$ the scaling function ϕ satisfies*

$$\sum_{n=-\infty}^{+\infty} n^k \phi(t-n) = \theta_k(t) \tag{4.21}$$

where θ_k is a polynomial of degree k.

It is desirable for the interval wavelets to have vanishing moments, such that we could, as on the whole line, have polynomial-to-zero and polynomial-to-polynomial mapping under high-pass and low-pass filtering respectively.

4.2.2 Meyer's Construction

In order to obtain interval wavelets that retain the same number of vanishing moments as the original wavelets defined on the whole line, one begins the construction

with scaling functions ϕ restricted to the half line $\mathbb{R}^+$:

$$\phi_{j,k}^{half}(x) = \begin{cases} 0 & \text{if } x < 0, \\ \phi_{j,k}(x) & \text{if } x \geq 0. \end{cases} \tag{4.22}$$

where support $\phi = [-N+1, N]$ and the spaces restricted to $[0, \infty)$,

$$V_j^{half} = \overline{\text{span}\{\phi_{j,k}^{half}; k \in \mathbb{Z}\}}, \tag{4.23}$$

form a multiresolution analysis for $\mathbf{L}^2(\mathbb{R}^+)$. Naturally, the corresponding wavelet spaces are given by

$$W_j^{half} = (V_j^{half})^{\perp} \cap V_{j-1}^{half}. \tag{4.24}$$

As mentioned in previous section, the problem is to construct properly the boundary-adapted scaling functions that can generate all polynomials on the interval, i.e., we want some functions $\phi_{j,k}^{edge}$ such that

$$\sum_j \sum_k a_{j,k} \phi_{j,k}^{edge}(t) = \theta_m(t) \mathbf{1}_{[0,N]}(t), \quad 0 \leq m < p, \tag{4.25}$$

where θ_m is some m^{th} degree polynomial. Observe that the the set of scaling functions $\phi_{j,k}^{half}$ defined in eqn. (4.22) is able to generate polynomials on the half line since

$$\sum_k a_k \phi_{j,k}^{half}(t) = \mathbf{1}_{[0,\infty)} \sum_k a_k \phi_{j,k}(t) = \mathbf{1}_{[0,\infty)} \theta_k(t). \tag{4.26}$$

From theorem 4.2.1, we see that the boundary wavelets constructed from these scaling functions should have appropriate vanishing moments. For the boundary and interior functions, $\{\phi_{j,k}^{half} | -2^{j-1}N+1 \leq k < 2^{j-1}N-1\}$ and $\{\phi_{j,k}^{half} | k \geq 2^{j-1}N-1\}$ respectively, Meyers [85] and Lemaieé *et al.* [90] had shown that

- The boundary functions are *independent*.
- The boundary functions are *orthogonal* to the interior functions.

These facts are exploited to construct orthonormal wavelet basis adapted to the half line. One begins by obtaining the orthonormal basis ϕ^{edge} of V^{half} by orthonormalizing the boundary functions:

$$\phi_{0,k}^{edge} = \sum_{n=-N+1}^{N-2} A_{k,n}\phi_{0,n}^{half}, \quad -N+1 \le k \le N-2. \tag{4.27}$$

where A is an invertible $(2N-2)\times(2N-2)$ matrix.

To obtain the boundary wavelets, we need to compute the projection of $\phi_{j-1,k}^{edge}$ onto W_j^{half} and orthonormalize them to yield the orthonormal basis ψ^{edge} of W^{half}. The projection is given by

$$\psi_{j,k}^{half} = P_{W_j^{half}}\phi_{j-1,k}^{edge} = \phi_{j-1,k}^{edge} - \sum_n \langle \phi_{j-1,k}^{edge}, \phi_{j,n}^{edge}\rangle \phi_{j,n}^{edge} \tag{4.28}$$

Finally, ψ^{edge} can be obtained by orthonormalizing ψ^{half}

$$\psi_{0,k}^{edge} = \sum_{n=-N+1}^{N-2} B_{k,n}\psi_{0,n}^{half} \tag{4.29}$$

The two orthonormalization matrices A and B can be obtained by Gram-Schmidt procedures or LU factorization.

With the edge functions properly defined, we continue to derive the refinement matrices, that would allow us to have fast dyadic decomposition with interval wavelets. By truncation, the two-scale relation for ϕ still holds for ϕ^{half}:

$$\phi_{0,k}^{half} = \sum_n h_n \phi_{-1,2k+n}^{half}, \quad k \ge -N+1, \tag{4.30}$$

Substituting ϕ^{half} for ϕ^{edge}, eqn. (4.27) can be rewritten to obtain a two-scale relation for the boundary functions:

$$\phi_{j,k}^{edge} = \sum_n H_{j-1,n}^{edge}\phi_{-1,n}^{edge} + \sum_m h_{k,m}^{edge}\phi_{-1,m}^{half} \tag{4.31}$$

where

$$H_{k.l}^{edge} = \sum_r \sum_m A_{k,r} h_m (A^{-1})_{2k+m,l} \tag{4.32}$$

$$h^{edge} = \sum_r A_{k,r} h_{m-2r} \tag{4.33}$$

and

$$\phi_j^{edge} = [\phi_{j,-N+1}^{edge}, \phi_{j,-N+2}^{edge}, \ldots]^T$$

$$\phi_j^{half} = [\phi_{j,-N+1}^{half}, \phi_{j,-N+2}^{half}, \ldots]^T$$

The new boundary low-pass filters are given by H^{edge} and h^{edge}. In matrix form, eqn. (4.31) can be written simply as

$$\phi_j^{edge} = AHA^{-1}\phi_{j-1}^{edge} \tag{4.34}$$

where $\phi_j^{half} = H\phi_{j-1}^{half}$ and the boundary filters are given by AHA^{-1}.

To obtain the boundary wavelet filters, eqn. (4.28) is rewritten in matrix form for simplicity:

$$\psi_j^{half} = \phi_{j-1}^{edge} - (\phi_{j-1}^{edge}(\phi_j^{edge})^t)\phi_j^{edge}, \tag{4.35}$$

From the dilation equation in eqn. (4.34), it follows that

$$\psi_j^{half} = \phi_{j-1}^{edge} - \phi_{j-1}^{edge}(\phi_{j-1}^{edge})^T (AHA^{-1})^T AHA^{-1}\phi_{j-1}^{edge} \tag{4.36}$$

Because of orthonormalization, we have $\phi_{j-1}^{edge}(\phi_{j-1}^{edge})^T = I$. Thus,

$$\psi_j^{half} = (I - (AHA^{-1})^T AHA^{-1})\phi_{j-1}^{edge} \tag{4.37}$$

Substituting into eqn. (4.29), we get the refinement matrix for boundary wavelet functions, or the high-pass boundary filters:

$$B(I - (AHA^{-1})^T AHA^{-1}) \tag{4.38}$$

Importantly, Meyer's construction still leads to filters in the framework of multiresolution hierarchy, which allows fast interval wavelet decomposition and reconstruction.

4.2.3 Cohen-Daubechies-Vial's Construction

In [86], Cohen *et al.* remarked on the stability problem of Meyer's boundary filters at higher order, which originates from the definition of ϕ^{half}. For large p, the energy of ϕ is largely concentrated on a fraction of its support. Thus, $\|\phi^{half}_{0,N-2}\|^2$ will be very close to 1 and $\|\phi^{half}_{0,-N+1}\|^2$, which likely to contain a small 'tail' of the original ϕ, will be very small (see figure 4.3). In effect, the matrix $\phi^{half}_j(\phi^{half}_j)^T$ can be very ill-conditioned and thus computation of the boundary filter coefficients requires greater precision. Meyer's boundary function has two other problems. Firstly, the ratio of scaling edge functions to wavelet edge functions is not 1 but $4p-2$ to $2p-1$. Secondly, the boundary functions tend to exhibit high oscillatory behavior near the borders, which suggested frequency leakage. In regard to these problems, Cohen *et al.* proposed a different spanning of the space V^{half}_j via binomial weights:

$$V^{half}_j = \overline{\text{span}\{T\phi^{half}_{j,k}; k \in \mathbb{Z}\}}, \tag{4.39}$$

where

$$T_{k,l} = \begin{cases} \begin{pmatrix} N-1-l \\ N-1-k \end{pmatrix} & \text{for } 0 \le k \le N-1, \\ \delta_{k,l} & \text{for } k \ge N. \end{cases} \tag{4.40}$$

Note that for Meyer's construction, $T = \mathbf{I}$. The calculation for the boundary filters proceeds as discussed in the last section.

4.2.4 Pre- and Post-conditioning Filters

For $\psi \in \mathbf{L^2}[\mathbb{R}]$ defined on the whole line with p vanishing moments, we have

$$\int_{-\infty}^{\infty} x^k \psi(x) = 0, \quad k = 0, \ldots, p-1. \tag{4.41}$$

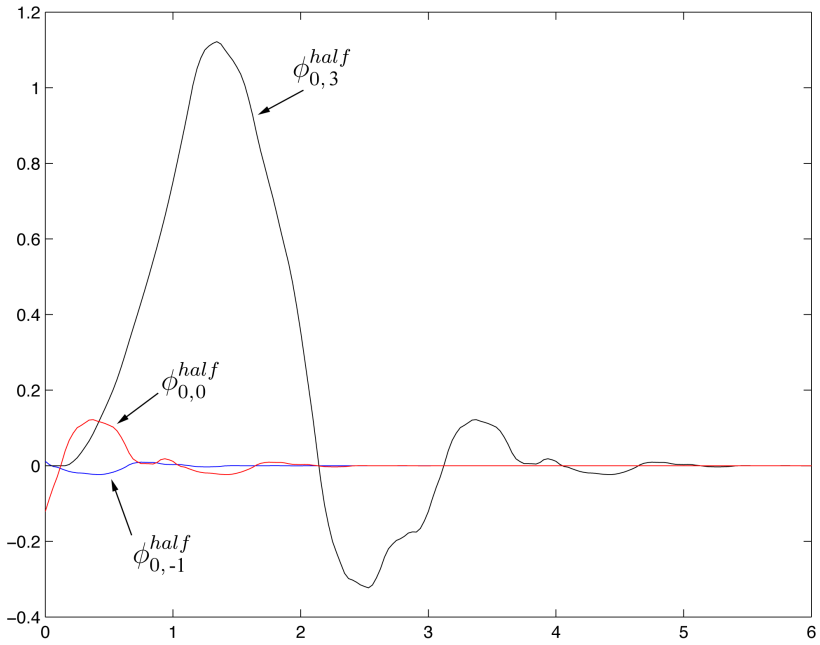


Figure 4.3: Tails of ϕ^{half} for $p = 4$.

.

Any polynomial sequence of degree up to $p-1$ will be mapped to zeros by the *high-pass* wavelet filter. However, as noted in [86], this is not necessarily true by default for wavelets defined on the interval. The original signal needs to be transformed, or pre-filtered, such that polynomial sequence would still be mapped to zeros during high-pass filtering. This procedure is important in applications where sparse representation is desired. Considering the half line $[0, \infty)$, the pre-filter is a $p \times p$ transformation matrix A such that

$$W = VA^T \tag{4.42}$$

where

$$\begin{aligned} (V_l^x)_k &= \int x^l \phi(x-k)dx & k,l &= 0,\ldots,p-1, \\ (W_l^x)_k &= \int x^l \phi_{0,k}^{left}(x-k)dx & k,l &= 0,\ldots,p-1 \end{aligned}$$

The post-filter, required for reconstruction, is simply A^{-1}. A different pre- and post-filter is defined similarly for the other half-line $(-\infty, 0]$.

For the purpose of continuity on this subject, we shall forward the discussion to

pre/post-conditioning in 2D images. In that scenario, we have three types of borders: horizontal, vertical and corners. If we should extend 1D decomposition steps to 2D by repeating row operations onto the columns, then we will have prefilters $A_{left}A_{left}^T$, $A_{right}A_{left}^T$, $A_{left}A_{right}^T$, and $A_{right}A_{right}^T$ on the four corners of the 2D image. With these 2D pre/post-conditioning filters, are simple polynomials (e.g. $\{1, x, y, xy\}$ for $p = 2$) still invariant under low pass filters and orthogonal to the high pass filter? Since we are using separable bases in 2D, tensor product gives:

$$\begin{aligned}(W^x)^T \otimes W^y &= (A(V^x)^T) \otimes (V^y A^T) \\ &= A((V^x)^T \otimes V^y)(A)^T\end{aligned} \tag{4.43}$$

$$\begin{aligned}(W^x)^T \otimes V^y &= (A(V^x)^T) \otimes (V^y) \\ &= A((V^x)^T \otimes V^y)\end{aligned} \tag{4.44}$$

$$\begin{aligned}(V^x)^T \otimes W^y &= (V^x)^T \otimes (V^y A^T) \\ &= ((V^x)^T \otimes V^y)A^T\end{aligned} \tag{4.45}$$

Thus the required 2D mappings are AA^T and A, which is obtained by separable pre-filtering the columns and rows of an image.

4.3 Proposed Alternate Interval Wavelet Designs

From the discussions in [86][87], we could have alternate constructions of the interval wavelets with a differently defined V_{half} space. For discussion purpose, we refer to the original filter constructed by Cohen *et al.* as Type-I. In the following we present two different sets of edge filters, Type-II and Type-III. We recall the definition of ϕ^{half} in eqn. (4.22) where it is assumed the support of ϕ be $[-N + 1, N]$. We can assume ϕ to have support on $[-N, N - 1]$ instead of $[-N + 1, N]$. This gives a shifted version of ϕ^{half} which leads to Type-II boundary filters with the same p vanishing moments as ψ. However, we have one less wavelet function at the boundary. We can also use a different

transformation matrix T found in eqn. (4.23):

$$T_{k,l} = \begin{cases} \begin{pmatrix} N-1-l \\ N-2-k \end{pmatrix} & \text{for } 0 \le k \le N-1, \\ \delta_{k,l} & \text{for } k \ge N. \end{cases} \tag{4.46}$$

This leads to Type-III boundary filters with one less scaling function at the boundary and we have $p' = p - 1$ vanishing moments.

4.4 General Boundary Filter Construction

Here we outline the general steps for computing Type-I, II and III boundary filters, and their respective pre- and post-conditioning filters. These procedures extend from the algorithm by Chyzak *et al.* [87].

1. Define parameters α and β,

$$(\alpha, \beta) = \begin{cases} (0,0) & \text{for Type-I filters,} \\ (0,1) & \text{for Type-II filters,} \\ (1,0) & \text{for Type-III filters.} \end{cases}$$

2. Define matrix $T := (T_{k,l}) \in \mathbb{R}^{N \times (2N-1)}$, with

$$(T_{k,l}) = \begin{pmatrix} N-l-1 \\ N-k-1-\alpha \end{pmatrix}$$

3. Define matrix $H := (H_{k,l}) \in \mathbb{R}^{(2N-1)\times(4N-2)}$, with

$$H_{k,l} = h_{l-2k+\beta} / \sqrt{2}$$

for $1 - N \le k \le N - 1$ and $1 - N \le l \le 3N - 2$.

4. Solve[1]

$$\Lambda = H \Lambda^{ext} H^t$$

[1] A recursive algorithm for solving Λ can be found in [87].

with $\Lambda \in \mathbb{R}^{(2N-1+\alpha)\times(2N-1+\alpha)}$ and

$$\Lambda^{ext} = \begin{pmatrix} \Lambda & 0 \\ 0 & I \end{pmatrix} \in \mathbb{R}^{(4N-2)\times(4N-2)}$$

5. Compute

$$\tilde{\Lambda} = T\Lambda T^t$$

6. Compute A by Cholesky Factorisation[2]

$$\tilde{\Lambda} = (A^{-1})(A^{-1})^t$$

7. Compute the dilation matrix for boundary scaling functions

$$H^{edge} = ATH(T^{ext})^{\dagger}(A^{ext})^{-1}$$

where $(T^{ext})^{\dagger}$ is the right inverse of T^{ext} and

$$T^{ext} = \begin{pmatrix} T & 0 \\ 0 & I \end{pmatrix} \in \mathbb{R}^{(3N-1+\alpha N+\alpha-\beta)\times(4N-2-\beta)}$$

and

$$A^{ext} = \begin{pmatrix} A & 0 \\ 0 & I \end{pmatrix} \in \mathbb{R}^{(3N-1+\alpha N+\alpha-\beta)\times(3N-1-\beta)}$$

8. Compute

$$C = (I - (H^{edge})^t H^{edge})$$

9. Define

$$G^{half} = (C_{k,l}) \in \mathbb{R}^{N\times(K+2N-1)}$$

10. Compute by unpivoted LU-decomposition, $U \in \mathbb{R}$, such that UG^{half} is lower triangular block matrix:

$$\tilde{C}^{-1} = LU$$

[2]Cholesky Factorization is only valid for orthogonal filters.

where

$$\tilde{C}_{k,l} = G^{half}_{k,N+2l}, \quad 0 \le k, l \le N-1.$$

11. Compute the dilation matrix G^{edge} of edge wavelet functions by orthogonalizing UG^{half} using Gram-Schmidt procedure.

12. Define

$$V = (T_{k,l})_{\substack{1 \le k \le N, \\ N \le l \le 2N-1}} \in \mathbb{R}^{N \times N}$$

13. The pre-conditioning matrix P is given by

$$P = (VA)^{-1}$$

14. The post-conditioning matrix Q is given by

$$Q = VA$$

Tables 4.3 and 4.4 list the filter coefficients for the case of Type-II boundary filters with $p' = p = 4$; and table 4.5 and 4.6 lists the coefficients for Type-III boundary filters with $p = 4$ and $p' = 3$. The plots of the boundary functions due to Type-I, II and III filters are found in figures 4.4, 4.5, 4.6 and 4.7. We have used the cascade algorithm

4.5 Numerical Examples

To compare the various approaches to perform wavelet transform on the interval, we consider the application of DWT on a ramp function using wavelets with 4 vanishing moments. Figure 4.8 shows a 2 level decomposition using Symlets with 4 vanishing moments and periodic extension. In figure 4.9, symmetric extension is applied but the *Cohen-Daubechies-Feauveau* (CDF) 9/7-tap biorthogonal wavelet [91] is used instead. For periodic extension, even though the ramp function is highly regular, the wavelet coefficients near the borders have magnitude of order 10^{-1}. The symmetric extension reduces the coefficient magnitude to order 10^{-3}. Examples of using interval wavelet

transform on the ramp function are given in figure 4.10 (Type-I filters) and 4.11 (Type-II and III filters). The wavelet coefficients generated by interval wavelet transform have remarkable magnitude order in the range of 10^{-12} to 10^{-11}, which are negligible in coding perspective. Border distortion can be observed in the approximated function, i.e., scaling coefficients, in both periodic and symmetric extension method. In contrast, the approximated function from the interval wavelet transform exhibit no such distortion — the approximated functions are still *ramp* functions themselves. The condition number of the interval wavelet matrices are also given in Table 4.2 as a comparison to extrapolated wavelet matrices.

4.6 Summary and Remarks

In this chapter, we have reviewed several techniques of overcoming the problem of performing discrete wavelet transform on finite functions. Though simple in application, the common methods such as periodic and symmetric extension have limitations where artificial singularities are generated at the signal borders. We also look at higher order extrapolation techniques like polynomial and wavelet extrapolation to reduce the effect of artificial singularities. We have revisited Meyer's work on the construction of wavelets living on the interval with vanishing moments. Using the construction guidelines by Cohen *et al.* [86], we have designed and proposed two additional sets of interval wavelet filters. These new filter sets differ from Cohen's construction in that, the number of scaling and wavelet functions are not equivalent. These filters can facilitate robust decomposition of arbitrary length sequences. In this chapter, we have also given a general matrix computation for the boundary filters and their corresponding preconditioning and postconditioning matrices. Numerical experiments on the newly proposed filters shows that with sufficient vanishing moments, they are able to map polynomials to zeros and polynomial to polynomial under high and low pass filtering.

Table 4.3: Type-II Left-Boundary Filter Coefficients, Symmlets ($p = 4$)

	l	$H_{k,l}$ or $h_{k,l}$	$G_{k,l}$ or $g_{k,l}$
$k = 0$	0	$8.410748699544336e-01$	
	1	$5.110372283456395e-01$	
	2	$-6.412061046851680e-02$	
	3	$-1.652953770905215e-01$	
$k = 1$	0	$-1.721709674782233e-01$	$3.012902656285995e-01$
	1	$5.203016052028449e-01$	$-5.227747587679865e-01$
	2	$6.927600124482274e-01$	$7.112311044202332e-01$
	3	$4.638037831940853e-01$	$-3.590800083529133e-01$
	4	$-2.474106621386358e-02$	$1.231321543066654e-02$
	5	$-6.325261317205616e-02$	$3.147976913394766e-02$
$k = 2$	0	$1.312560618065118e-01$	$3.901724930498521e-01$
	1	$-1.165778158313018e-01$	$-4.192406144795308e-01$
	2	$-1.774736115980857e-02$	$-9.617178402286998e-02$
	3	$3.143375439267822e-01$	$7.264771272684965e-01$
	4	$7.932651172198317e-01$	$-3.511743237843509e-01$
	5	$4.840265999152468e-01$	$-9.961073615251531e-02$
	6	$-2.938460767973613e-02$	$1.519063161063818e-02$
	7	$-7.512421682690058e-02$	$3.883612520177068e-02$
$k = 3$	0	$-2.048689042467881e-02$	$-4.787437888913756e-02$
	1	$2.912555754411917e-02$	$6.806147506340154e-02$
	2	$9.297131473066246e-03$	$2.172581520084711e-02$
	3	$-1.780389927239799e-02$	$-4.160468496694576e-02$
	4	$-9.322680035945163e-02$	$-4.808048506677421e-01$
	5	$3.030865705537470e-01$	$8.111137337927663e-01$
	6	$8.027938065526461e-01$	$-3.003002926245245e-01$
	7	$4.966808051256452e-01$	$-1.019995745023441e-01$
	8	$-2.960740010478690e-02$	$1.266989931426074e-02$
	9	$-7.569380436842242e-02$	$3.239166143150159e-02$

The corresponding pre-conditioning matrix $A_{left,pre}$ is as follows:

$$\begin{bmatrix} 4.588010558\text{e-}01 & 3.476532653\text{e-}01 & -3.053740436\text{e-}01 & 6.824705749\text{e-}02 \\ 0 & 1.197827425\text{e+}00 & -1.495414184\text{e-}01 & 3.974887730\text{e-}02 \\ 0 & 0 & 1.008539163\text{e+}00 & -6.203155081\text{e-}03 \\ 0 & 0 & 0 & 1.000950017\text{e+}00 \end{bmatrix}$$

The post-conditioning matrix $A_{right,post}$ are obtained by taking the corresponding inverse of $A_{right,pre}$.

Table 4.4: Type-II Right-Boundary Filter Coefficients, Symmlets ($p = 4$)

	l	$H_{k,l}$ or $h_{k,l}$	$G_{k,l}$ or $g_{k,l}$
$k = -1$	-1	$-5.934437509312862e-02$	
	-2	$2.945622329344726e-01$	
	-3	$-2.072054233150544e-01$	
	-4	$9.310087263921459e-01$	
$k = -2$	-1	$4.312269051690282e-01$	$-7.957093088484318e-01$
	-2	$8.035066271961139e-01$	$1.580546024113178e-01$
	-3	$3.782790792877906e-01$	$5.839042743022116e-01$
	-4	$-1.425446771340545e-01$	$2.922673360503910e-02$
	-5	$-2.577443994080112e-02$	$2.982474497954964e-03$
	-6	$6.589451987286582e-02$	$-7.624946479038657e-03$
$k = -3$	-1	$3.751270796989729e-01$	$1.821667196224223e-01$
	-2	$-4.041754924467591e-01$	$-2.814874413829827e-01$
	-3	$6.015792351176238e-01$	$3.255178140662214e-01$
	-4	$2.856761598571271e-01$	$1.731188884923203e-01$
	-5	$4.662239297164217e-01$	$-7.311183875515722e-01$
	-6	$-1.795864884204323e-01$	$4.601029655775826e-01$
	-7	$-1.926632114088405e-02$	$2.681616171960735e-02$
	-8	$4.925596770330917e-02$	$-6.855776906909311e-02$
$k = -4$	-1	$-5.414695871364911e-02$	$-2.305134308246142e-02$
	-2	$5.514825199520724e-03$	$2.347761337780246e-03$
	-2	$-6.613942153832640e-02$	$-2.815675224094761e-02$
	-2	$-1.991627249601784e-02$	$-8.478718434867647e-03$
	-2	$4.817318359908384e-01$	$9.264772134446057e-02$
	-2	$8.121312070272604e-01$	$3.017606747181544e-01$
	-2	$2.987523844325451e-01$	$-8.033735663859395e-01$
	-2	$-1.009543480132192e-01$	$4.969200433708864e-01$
	-2	$-1.261447795057246e-02$	$2.963105525669948e-02$
	-2	$3.224997206181687e-02$	$-7.575428075066273e-02$

The corresponding pre-conditioning matrix $A_{right,pre}$ is as follows:

$$\begin{bmatrix} 3.458211574\text{e-}02 & -3.161200498\text{e-}02 & 7.042820412\text{e-}01 & -2.465787046\text{e-}02 \\ 0 & 4.890103254\text{e-}01 & 2.771911480\text{e-}001 & -2.575918993\text{e-}02 \\ 0 & 0 & 6.541968843\text{e-}01 & 3.351242307\text{e-}02 \\ 0 & 0 & 0 & 9.991667758\text{e-}01 \end{bmatrix}$$

The post-conditioning matrix $A_{right,post}$ are obtained by taking the corresponding inverse of $A_{right,pre}$.

Table 4.5: Type-III Left-Boundary Filter Coefficients, Symmlets ($p' = 3, p = 4$)

	l	$H_{k,l}$ or $h_{k,l}$	$G_{k,l}$ or $g_{k,l}$
$k = 0$	0		$-7.977472501504831e-02$
	1		$4.378784742493808e-01$
	2		$-8.015059975704307e-01$
	3		$3.993576979306099e-01$
$k = 1$	0	$9.062312766267855e-01$	$-1.911540970932423e-01$
	1	$3.910933368252937e-01$	$5.738874002341042e-01$
	2	$1.503692152955543e-01$	$-4.256783585142284e-02$
	3	$5.399877176744022e-02$	$-7.528603606065529e-01$
	4	$-5.919921115565505e-03$	$9.322993621308390e-02$
	5	$-1.513477540115587e-02$	$2.383501600286601e-01$
$k = 2$	0	$-2.818110272623821e-01$	$-1.658463680540387e-01$
	1	$4.108299846528714e-01$	$2.597103026595094e-01$
	2	$4.963260218384510e-01$	$2.449520379342258e-01$
	3	$4.893690047912579e-01$	$1.737255307696053e-01$
	4	$4.528831304602435e-01$	$-8.761510093798820e-01$
	5	$2.420485069711628e-01$	$1.768614857913477e-01$
	6	$-1.742849856259770e-02$	$4.599490870941610e-02$
	7	$-4.455741996810146e-02$	$1.175898461017079e-01$
$k = 3$	0	$8.267433721168246e-02$	$1.486050428513191e-01$
	1	$-1.456310582446918e-01$	$-2.617681661500549e-01$
	2	$-7.962439420304909e-02$	$-1.431228469065717e-01$
	3	$1.638781130707737e-02$	$2.945667882817803e-02$
	4	$1.061456705291942e-01$	$-2.199402884226711e-02$
	5	$4.129710212020809e-01$	$8.255364837205581e-01$
	6	$7.559714101898580e-01$	$-4.022938165154847e-01$
	7	$4.539544437125418e-01$	$-2.058160717774155e-01$
	8	$-2.814239631596661e-02$	$1.565662860102441e-02$
	9	$-7.194839917250885e-02$	$4.002748563536959e-02$

Type-III edge filters are calculated using the algorithm by Chyzak *et al.* with matrix $(T_{k,l}) = \binom{N-l-1}{N-k}$. The corresponding pre-conditioning matrix $A_{left,pre}$ is as follows:

$$\begin{bmatrix} 5.0060679978e+00 & -5.684780553e+00 & 2.026117054e+00 & -3.999321888e-16 \\ 0 & 1.700406236e+00 & -4.394123865e-01 & 5.795999714e-14 \\ 0 & 0 & 1.053056297e+00 & -3.038575389e-13 \\ 0 & 0 & 0 & 1.000000000e+00 \end{bmatrix}$$

The post-conditioning matrix $A_{right,post}$ are obtained by taking the corresponding inverse of $A_{right,pre}$.

Table 4.6: Type-III Right-Boundary Filter Coefficients, Symmlets ($p' = 3, p = 4$)

	l	$H_{k,l}$ or $h_{k,l}$	$G_{k,l}$ or $g_{k,l}$
$k = -1$	-1		$-1.458533252931872e-01$
	-2		$5.769372496146517e-01$
	-3		$-7.468926950076344e-01$
	-4		$2.966842086330747e-01$
$k = -2$	-1	$9.079399472964419e-01$	$-2.879283444897474e-01$
	-2	$3.981455669229031e-01$	$5.689728381967509e-01$
	-3	$1.297468662541737e-01$	$2.253807529894668e-01$
	-4	$-1.253727805054632e-03$	$-6.805946760422796e-01$
	-5	$-6.196253401940596e-03$	$-1.026198102907073e-01$
	-6	$1.584124209368918e-02$	$2.623561615331947e-01$
$k = -3$	-1	$-2.604509085817922e-01$	$9.831099468128711e-03$
	-2	$4.070446338783698e-01$	$3.978021956087043e-02$
	-3	$5.986048813914320e-01$	$-9.554383465861972e-02$
	-4	$5.873813482611784e-01$	$-3.130526713952643e-01$
	-5	$2.410561066658228e-01$	$8.015781293577223e-01$
	-6	$-6.478524456109926e-02$	$-4.919787512696180e-01$
	-7	$-1.049561424625316e-02$	$-2.963773723862898e-02$
	-8	$2.683291909023560e-02$	$7.577136379852195e-02$
$k = -4$	-1	$5.487898585654069e-02$	$-2.293684059779154e-02$
	-2	$-1.224930985002266e-01$	$5.119636631081655e-02$
	-3	$-7.754771409843071e-02$	$3.241130504263837e-02$
	-4	$6.995752433997030e-02$	$-2.923896189410875e-02$
	-5	$5.204843297205952e-01$	$-1.054159746883348e-01$
	-6	$7.775649112003379e-01$	$-2.811294440844721e-01$
	-7	$2.922891257723205e-01$	$8.058107008525132e-01$
	-8	$-9.470843837187934e-02$	$-4.988152308552150e-01$
	-9	$-1.241887608351350e-02$	$-2.971356603333534e-02$
	-10	$3.174989946486852e-02$	$7.596522647919292e-02$

Type-III edge filters are calculated using the algorithm by Chyzak *et al.* with matrix $(T_{k,l}) = \binom{N-l-1}{N-k}$. The corresponding pre-conditioning matrix $A_{right,pre}$ is as follows:

$$\begin{bmatrix} 2.034127148e+00 & -1.352184118e+00 & 4.482866111e-01 & -9.842490591e-16 \\ 0 & 1.200879431e+00 & -1.155136617e-01 & 8.285443053e-14 \\ 0 & 0 & 1.014904020e+00 & -3.312954890e-13 \\ 0 & 0 & 0 & 1.000000000e+00 \end{bmatrix}$$

The post-conditioning matrix $A_{right,post}$ are obtained by taking the corresponding inverse of $A_{right,pre}$.

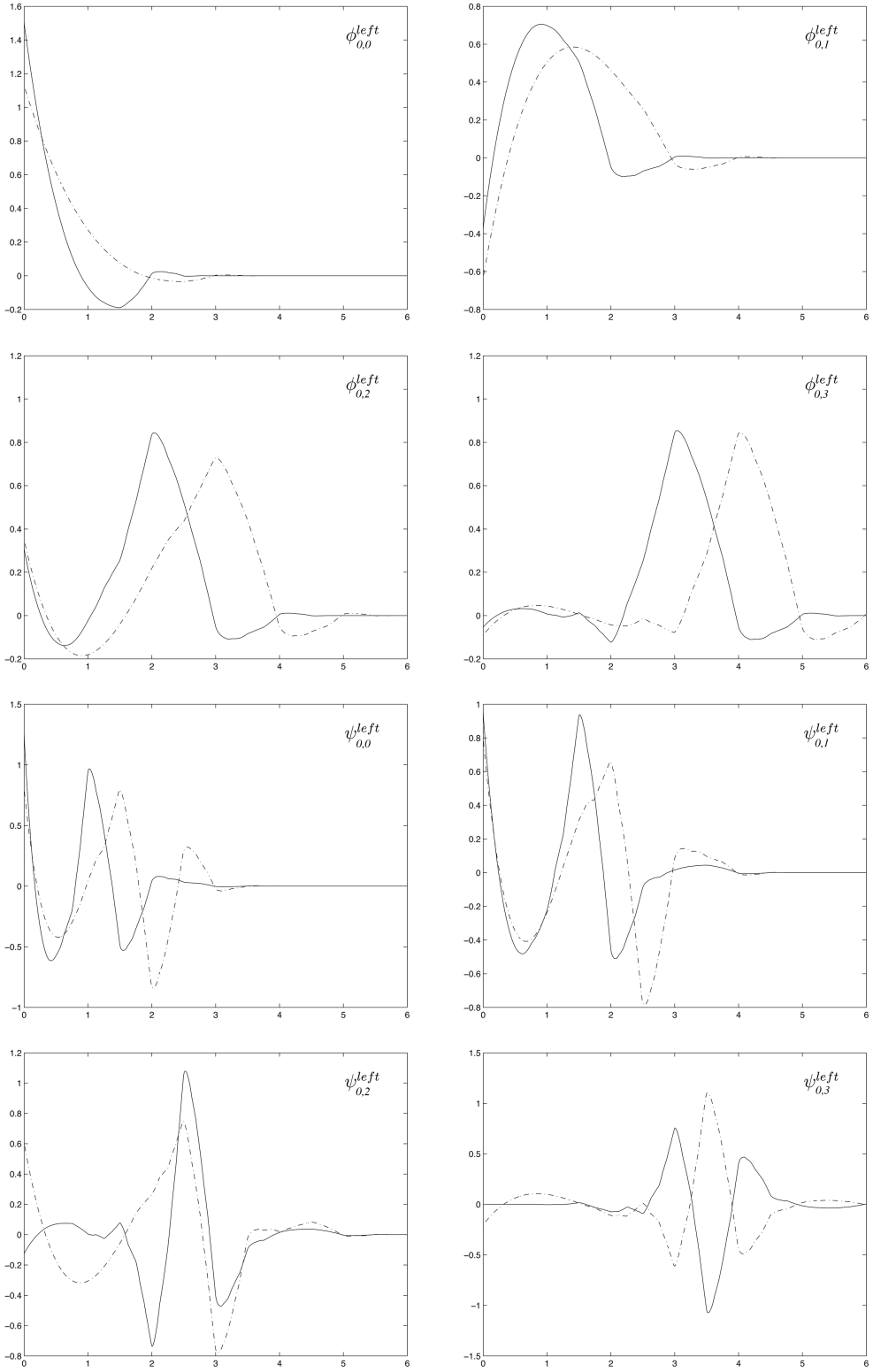


Figure 4.4: Left boundary scaling ϕ^{left} and wavelet ψ^{left} functions (symmlets, $p = 4$). (*s*olid: Type-II, *d*ashed: Type-I)

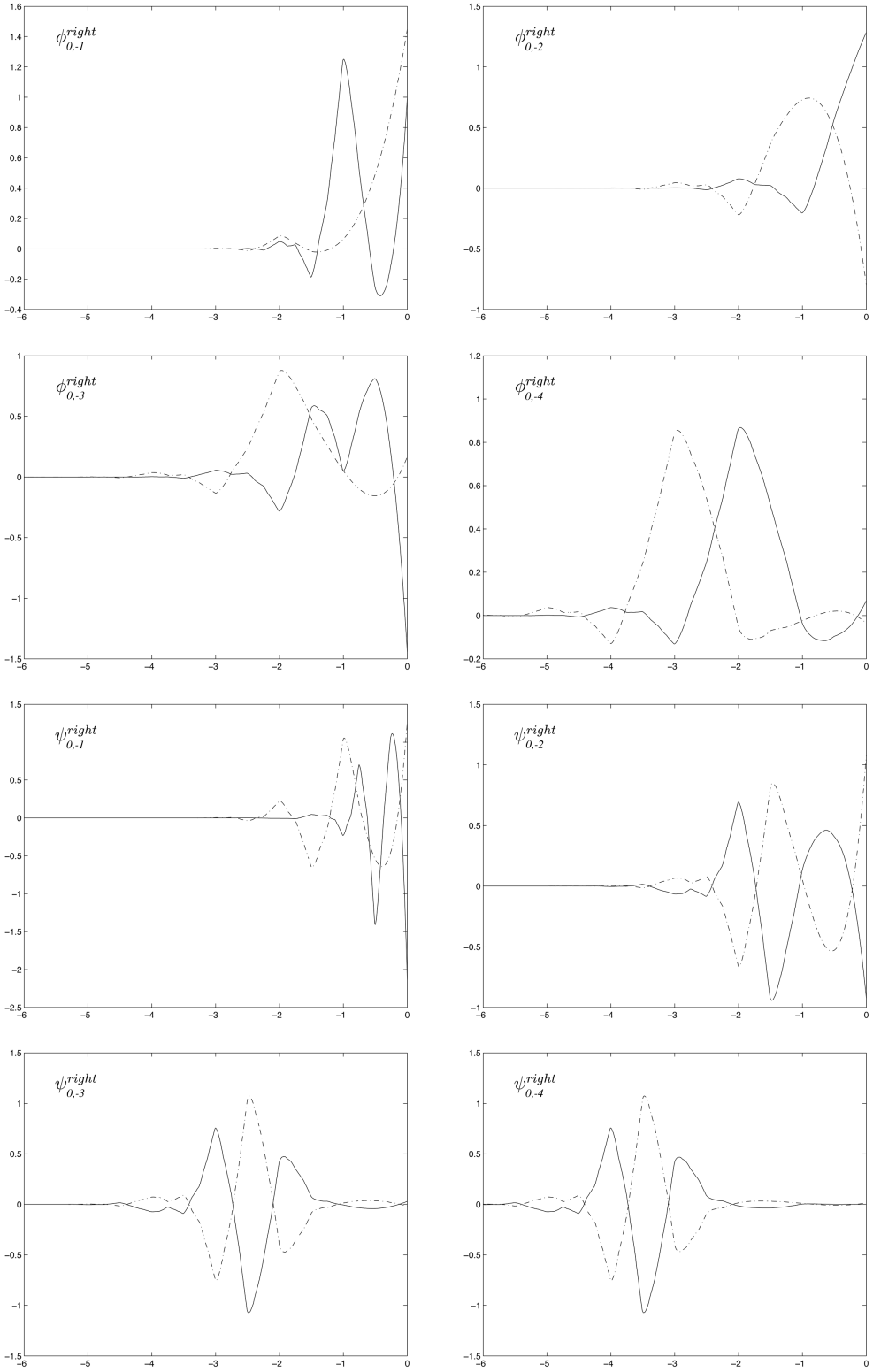


Figure 4.5: Right boundary scaling ϕ^{right} and wavelet ψ^{right} functions (symmlets, $p = 4$). Note that Type-II has one less wavelet function. (*s*olid: Type-II, *d*ashed: Type-I)

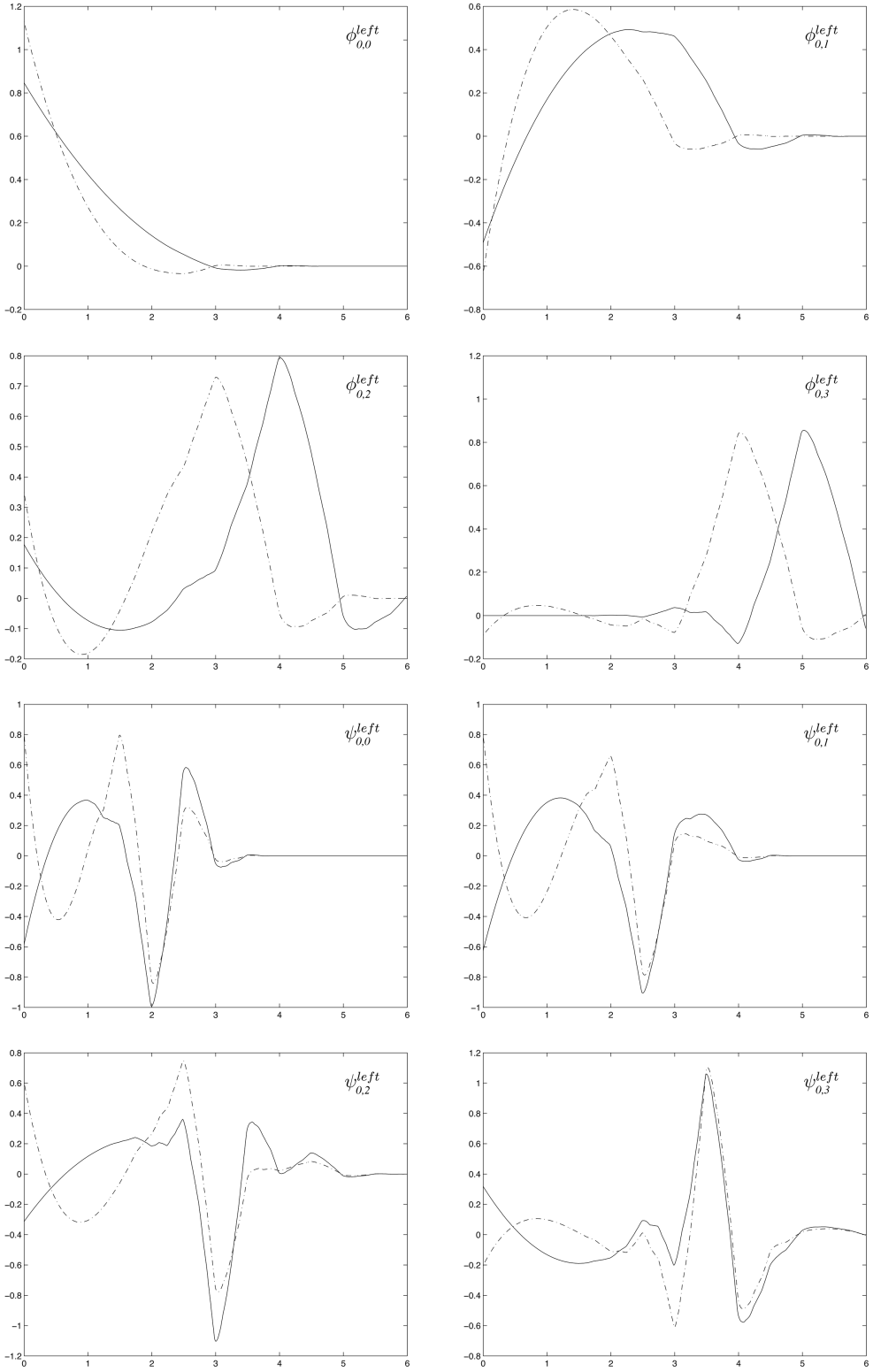


Figure 4.6: Left boundary scaling ϕ^{left} and wavelet ψ^{left} functions (symmlets, $p = 4$). (*s*olid: Type-III, *d*ashed: Type-I)

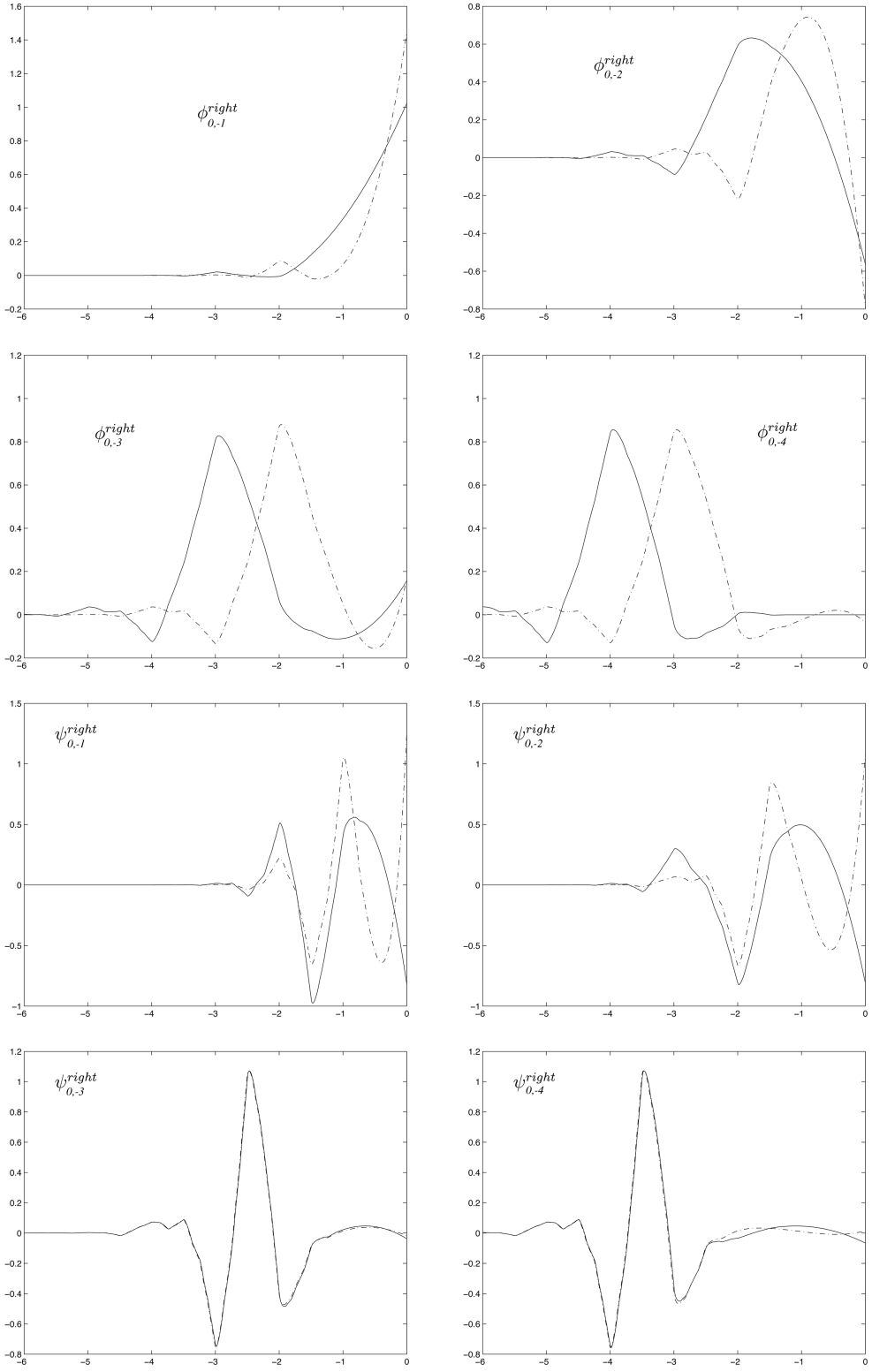


Figure 4.7: Right boundary scaling ϕ^{right} and wavelet ψ^{right} functions (symmlets, $p = 4$). (*s*olid: Type-III, *d*ashed: Type-I)

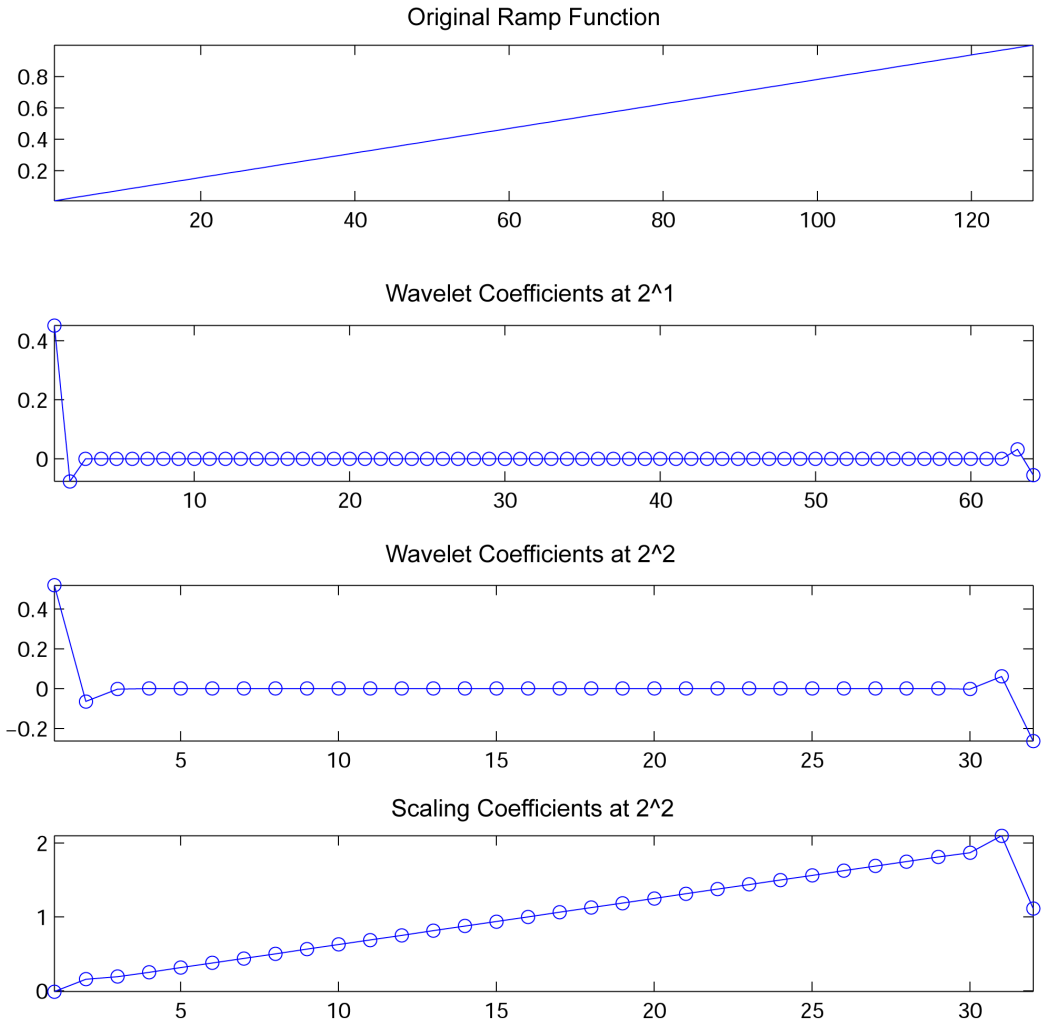


Figure 4.8: Periodic extension example using *ramp* function.

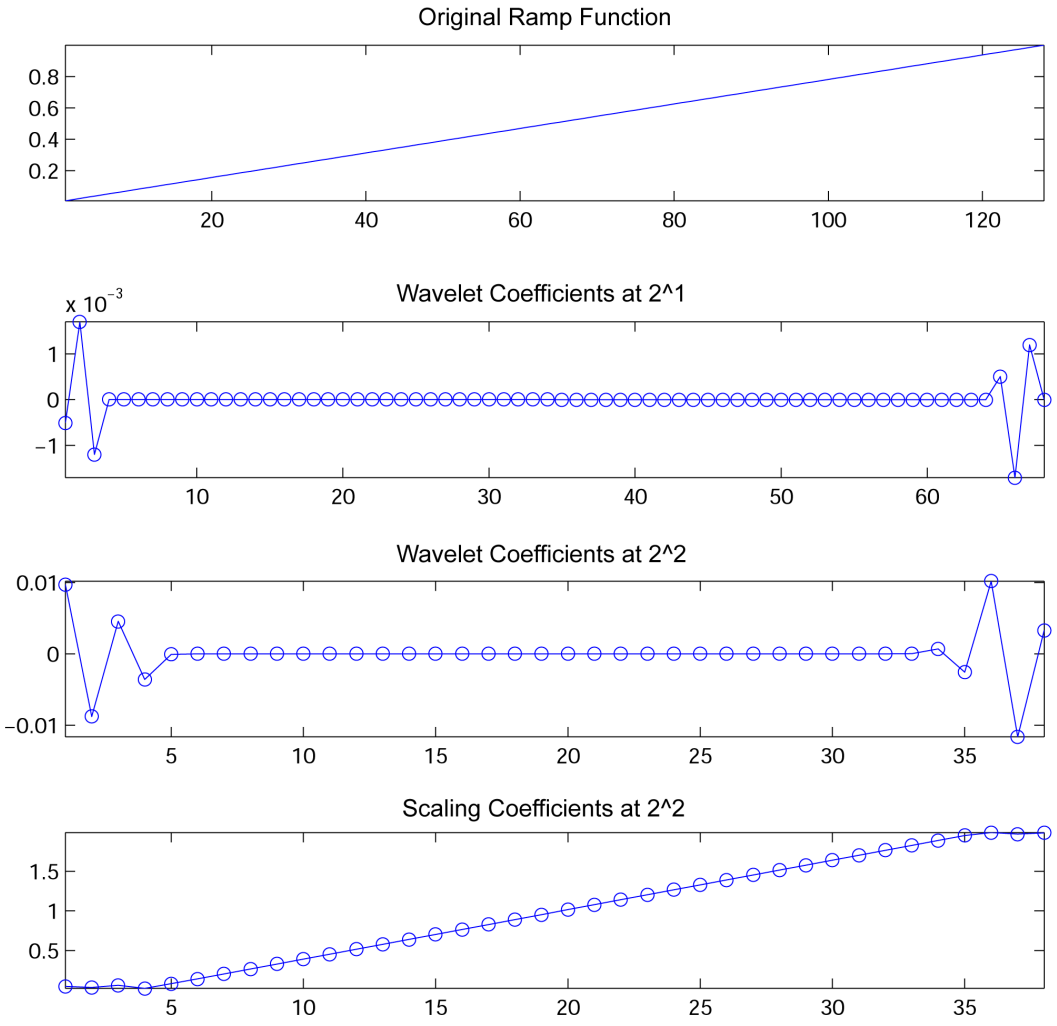


Figure 4.9: Symmetric extension example using *ramp* function.

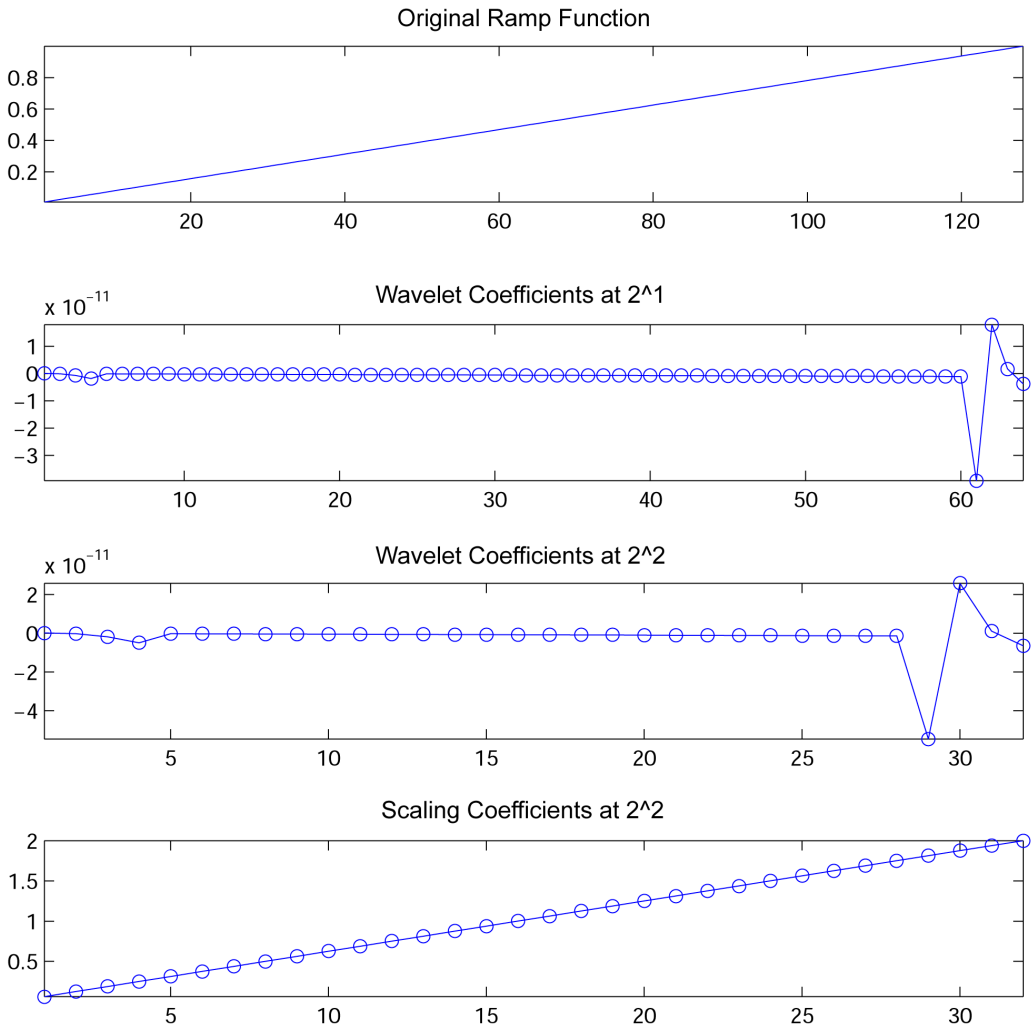


Figure 4.10: Type-I (for left and right border) interval wavelets example using *ramp* function.

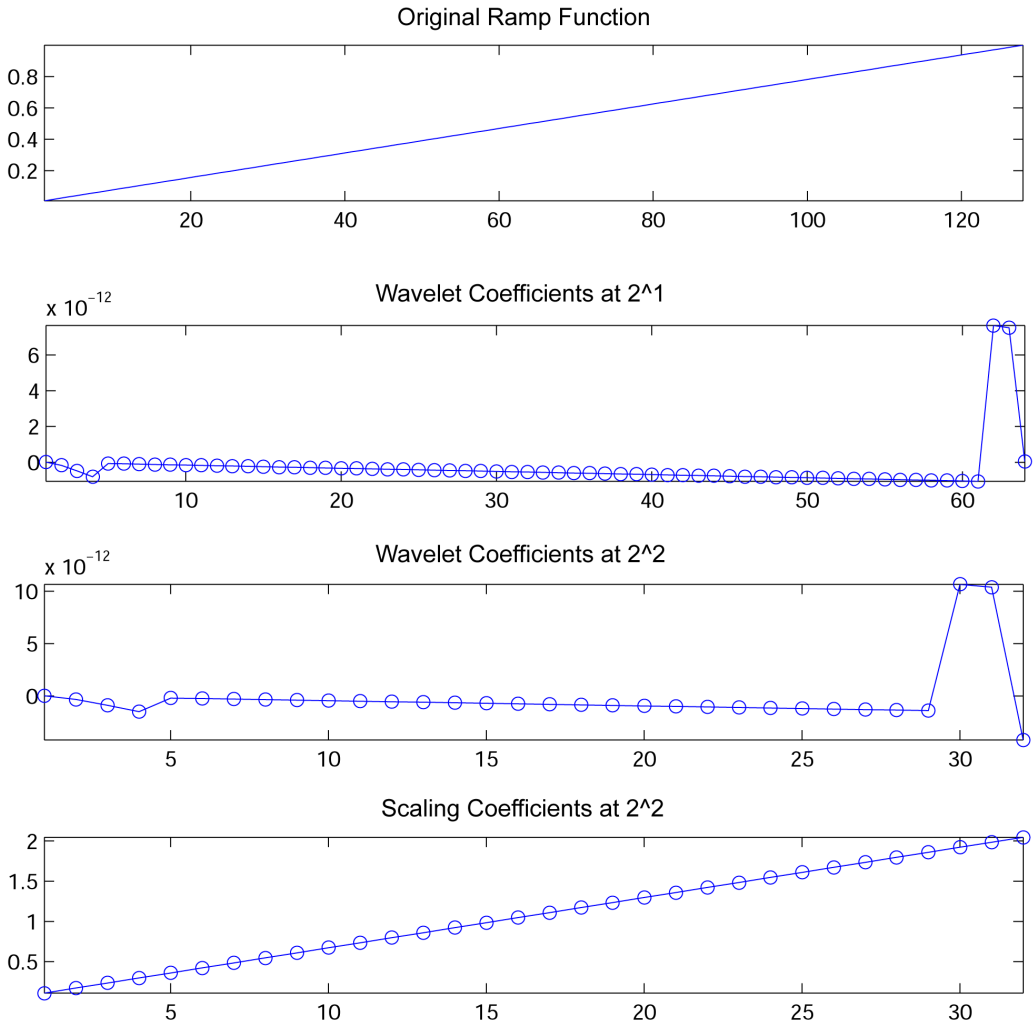


Figure 4.11: Type-II (for left border) and III (for right border) interval wavelets example using *ramp* function.

5 Signal Singularities: Detection, Analysis and Synthesis

Singularity is almost invariably a clue. —Arthur Conan Doyle Sr. (1859-1930).

Often, some of the most interesting and important information in a signal is found in irregular structures and transient phenomena. For example, discontinuities in image intensity usually indicate the presence of object edges. If a signal f has a singularity at v, then f is not differentiable at v and Lipschitz exponent can be used to characterize this singularity behavior. Wavelets are known for their ability to generate sparse representations for smooth functions. At the singularities, a finite number of wavelet coefficients are generated, which in contrast, is far less than Fourier series representation. This ability of wavelets to efficiently approximate particular classes of functions with few non-zero wavelet coefficients has been exploited in applications such as data compression, noise removal and fast computing. For a function f to have few non-negligible wavelet coefficients depends mostly on the regularity of f, the vanishing moments of ψ and its support size. In this chapter, we study the relationship between signal regularity and wavelet representation and the problems posed by singularities in the context of image

compression using standard wavelet transform. We will also propose the application of interval wavelets for efficient piecewise signal analysis and synthesis.

5.1 Signal Regularity and Lipschitz Exponent

The Lipschitz exponent is a measure of how smooth or regular a signal is at a certain point. It is an extension of how many times the signal is differentiable at that point. If a signal is regular, it can be locally approximated by a polynomial. The Taylor formula relates the differentiability of a signal to polynomial approximation. Suppose f is m times differentiable in $[v-h, v+h]$ and p_v is the Taylor polynomial expansion in the neighborhood of v:

$$p_v = \sum_{k=0}^{m-1} \frac{f^k(v)}{k!}(t-v)^k \tag{5.1}$$

The Taylor formula shows that the approximation error $e_v(t) = f(t) - p_v(t)$ satisfies

$$\forall t \in [v-h, v+h], \quad |e_v| \leq \frac{|t-v|^m}{m!} \sup_{u \in [v-h, v+h]} |f^m(u)| \tag{5.2}$$

When $t \to v$, the m^{th} order differentiability of f in neighborhood of v gives an upper bound on $e_v(t)$. The Lipschitz regularity refines this upper bound with non-integer exponents. If f has a singularity at v, it is not differentiable at v and the Lipschitz exponent at v will characterize this singular behavior. The definition of Lipschitz regularity is given by the following.

Definition 5.1.1. *(Lipschitz)*

- *A function f is pointwise Lipschitz $\alpha \geq 0$ at v, if there exists $K > 0$, and a polynomial p_v of degree $m = \lfloor \alpha \rfloor$ such that*

$$\forall t \in \mathbb{R}, \quad |f(t) - p_v(t)| \leq K|t-v|^\alpha \tag{5.3}$$

  where α is also called the Lipschitz exponent.

- *A function f is uniformly Lipschitz α over $[a, b]$ if it satisfies the Lipschitz condition 5.3 for all $v \in [a, b]$ with a constant K independent of v.*

- *The Lipschitz regularity of f at v or over $[a, b]$ is the supremum of α such that f is Lipschitz α.*

To give a few examples, a signal that is differentiable once has Lipschitz regularity 1, a step function is Lipschitz 0 and a dirac impulse is Lipschitz -1. Thus, if the Lipschitz regularity is $\alpha < 1$ at v, then f is not differentiable at v and α characterizes the nature of the singularity.

Uniform Lipschitz exponents provide a global regularity measure. Hence, f is necessarily m times differentiable near v if it is uniformly Lipschitz $\alpha > m$ in the neighborhood of v. The uniform Lipschitz regularity of f over $\mathbb{R}$ can be related to the asymptotic decay of its Fourier transform.

Theorem 5.1.1. *A function f is bounded and uniformly Lipschitz α over $\mathbb{R}$ if*

$$\int_{-\infty}^{+\infty} |\hat{f}(\omega)|(1 + |\omega|^{\alpha})d\omega < +\infty. \tag{5.4}$$

The pointwise regularity of f cannot be measured from the decay of $|\hat{f}(\omega)|$ at high frequencies ω. However, it is possible to measure the local regularity of f by using the wavelet transform since wavelets are well localized in time, and most importantly, due to the vanishing moments of wavelets.

5.2 Wavelets and Singularities

Recall ψ has p vanishing moments if

$$\int_{-\infty}^{+\infty} t^k \psi(t)dt = 0, \quad \text{for } 0 \leq k < p. \tag{5.5}$$

If f is regular and ψ has enough vanishing moments then the wavelet coefficients $|\langle f, \psi_{j,n}\rangle|$ are small at fine scales 2^j. The Lipschitz condition, eqn. (5.3), approximates f with a polynomial p_v in the neighborhood of v with an error e_v:

$$f(t) = p_v(t) + e_v(t) \tag{5.6}$$

$$e_v(t) \leq K|t - v|^{\alpha} \tag{5.7}$$

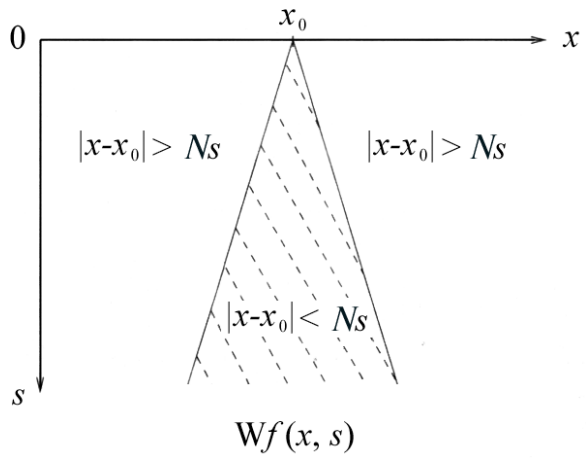


Figure 5.1: The cone of influence of some wavelet, with support $[-N, N]$ at x_0 is given by the shaded region. The cone of influence highlights the set of wavelets coefficients, $\langle f, \psi_{u,s} \rangle$ that depends on the value of f in the neighborhood of v. Conversely, the group of wavelet coefficients under the same cone of influence contributes towards the fine scale feature at v.

A wavelet with p vanishing moments is orthogonal to polynomials of degree up to $p-1$. Since $\alpha < p$, the polynomial p_v has at most degree $p-1$. We can verify easily that the wavelet operator on p_v gives:

$$Wp_v(u,s) = \int_{-\infty}^{+\infty} p_v(t) \frac{1}{\sqrt{s}} \psi(\frac{t-u}{s}) dt = 0 \tag{5.8}$$

The wavelet transform ignores the polynomials and only approximates the error function e_v.

Like Fourier coefficients, the wavelets coefficients decay rapidly for smooth functions. For Fourier, f has to be smooth everywhere. A small jump will make the coefficients decay no faster than order -1. For wavelets, the slow decay only applies to coefficients in the neighborhood of the jump. Consider the *cone of influence* for a point v. The cone of influence of v in the scale-space plane is the set of points (u, s) (see figure 5.1) such that v is included in the support of $\psi_{u,s}(t) = \frac{1}{\sqrt{s}} \psi(\frac{t-u}{s})$. If the signal is uniformly Lipschitz α in the neighborhood of v, then wavelet coefficients in the cone of influence have a decay that is related to the Lipschitz regularity α by virtue of $Wf(u,s) = We_v(u,s)$ and eqn. (5.7).

Theorem 5.2.1. *Assume p vanishing moments wavelet. If $f \in L^2(\mathbb{R})$ is uniformly Lipschitz*

$\alpha \leq p$ over $[a, b]$ then there exists $A > 0$ such that

$$\forall (u, s) \in [a, b] \times \mathbb{R}^+, \quad |Wf(u, s)| \leq A s^{\alpha + \frac{1}{2}} \tag{5.9}$$

Conversely, if $Wf(u, s)$ satisfies (5.9) and if $\alpha \leq n$ is not an integer, then f is uniformly Lipschitz α on $[a + \epsilon, b - \epsilon]$ for any $\epsilon > 0$.

When the scale s decreases, $Wf(u, s)$ measures the fine scale variations in the neighborhood of u. Essentially, theorem 5.2.1 shows that $|Wf(u, s)|$ decays over the intervals where f is uniformly Lipschitz α and how this decay is being related to the Lipschitz regularity α . This relation could certainly be exploited to detect different kinds of singularities within a signal.

5.3 Detecting and Characterizing Singularities

In previous sections, the basic theorems that relate the local Lipschitz exponents of a function to the evolution across scales of the wavelet coefficients was discussed. In practice, these theorems do not provide simple and direct strategies for detecting and characterizing the singularities of a signal. The following sections show that the wavelet transform local maxima provide an efficient approach for studying and detecting irregularities in signals.

5.3.1 Wavelet Modulus Maximas

Wavelet maximas [92][73] are wavelet coefficients that are strict local maximums on the decomposition map. These maximas contain information about signal singularities which could be used for edge detection in multi-scale schemes. The propagation of these maximas across scales, or *maxima lines*, are always in the *cone of influence* of some abscissa at fine scale that gives the location of signal discontinuities or image edges. First, we define what is meant by local maximum:

- The *local maxima* of the wavelet transform of $f(x)$, any point (u_0, s_0) on the space-scale map such that when u belongs to either a right or the left neighborhood of u_0, $|Wf(u, s_0)| < |Wf(u_0, s_0)|$ and when u belongs to the other side of the neighborhood of u_0, $|Wf(u, s_0)| \leq |Wf(u_0, s_0)|$.

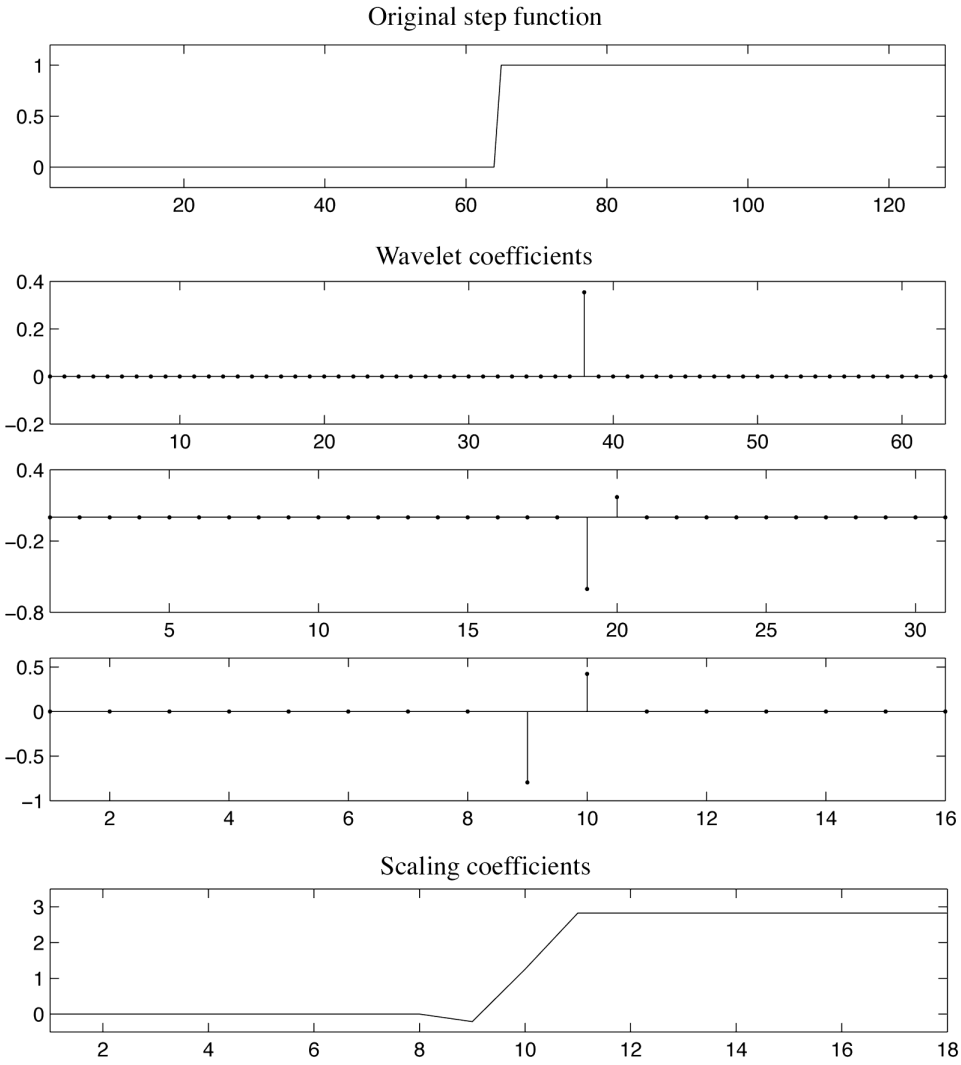


Figure 5.2: Step function and cone of wavelet coefficients. Decomposition performed at 3 levels with Daubechies $p = 2$ wavelets.

- The *maxima line* of the wavelet transform is any connected curve in the space-scale (u, s) along which all points are local maxima of the wavelet transform.

We give a brief introduction of the calculation of wavelet transform maxima. Supposed that wavelet ψ^1 has p vanishing moments and a compact support. Then there exists a function θ of compact support such that its partial differentiation gives

$$\psi = (-1)^p \frac{\partial^p \theta}{\partial x^p}$$

and

$$\int_{-\infty}^{+\infty} \theta(t)dt \neq 0$$

The dyadic wavelet transform can be written as a multiscale differential operator:

$$Wf(u,s) = \langle f(x), \psi_j(x-u) \rangle = s^p \frac{\partial^p d}{\partial u^p} f \star \bar{\theta}_s(u) \tag{5.10}$$

where $\bar{\theta}_s(u) = \frac{1}{\sqrt{s}}\theta)(-\frac{u}{s})$. It has been proven in [93] that if $Wf(u,s)$ has no modulus maxima at fine scales, then f is locally regular. It can be implied that f can be singular at point v only if there is a sequence of wavelet maxima points that converges towards v at fine scales. Thus it is guaranteed that all singularities can be detected by following wavelet transform modulus maxima at fine scales.

5.3.2 Multiscale Detection

Wavelet modulus maxima at fine scale is sufficient to give us locations of singularities in the signals. However, we are not interested in all singularities. For example noises in signals also contributes to modulus maxima in the fine scales. Noise has always been a problem in feature extraction applications such as image edge detection. Since, an edge is defined simply as discontinuity, then noise falls under this category too. Even though proper thresholding could eliminate some of the 'weak' edges and noise, detection by solely spatial differences is usually too sensitive to noise, thus producing unsatisfactory result in the absence of human intervention. Thus, it is necessary to discriminate the

singularities through some form of characterization. By rewriting eqn. (5.9) as

$$\log_2 |Wf(u,s)| \leq \log_2 A + (\alpha + \frac{1}{2}) \log_2 s, \tag{5.11}$$

we can obtain the Lipschitz regularity α at u by measuring the maximum slope of $\log_2 |Wf(u,s)|$ as a function of $\log_2 s$. This Lipschitz regularity measurement would allow us to discriminate detected singularities. For example, if we are only interested in step edges, we would choose singularities whose regularity is positive and near 0. For smoothed singularities, the decay of $log_2|Wf(u,s)|$ increases, leading to larger α. Sharp irregularities can be filtered out since their measured Lipschitz exponents would be negative.

5.4 Analysis and Synthesis of Singularities

In this section, we look at the problems of reconstructing signals in the neighborhood of their discontinuities and the distortions that are unique to wavelet decompositions. In lossy compression, discarding or quantizing the wavelet coefficients will inevitably lead to signal distortions. The general assumption is that higher frequency or finer scale components can be discarded without causing much perceptible distortion since the HVS is predominantly sensitive to mid-to-low range components. Such q model of the HVS can be overly simplistic and ideal. In the following, we give a brief review of two existing methods, *wavelet footprints* and *ENO wavelets*, which succumb to these distortions, and propose a new technique based on interval wavelets. Though the following discussion is generally confined to one dimensional domain, the arguments apply equally to two dimensional problems such as image compression, which is discussed in subsequent chapters.

5.4.1 Quantization Distortion

In compression, a signal is transformed and represented by a finite number of basis functions. The nature of distortion or compression artifacts depends on the transform basis. For example, in Fourier transform, the use of finite series to approximate discontinuous waveforms produces *Gibbs* phenomenon. In wavelet transform, discarding

coefficients that belong to the cone of influence of some singularities or signal discontinuities leads to distortions such as *blurring* and *ringing*.

Blurring is the result of interpolation and thus, loss of fine scale details. In wavelet transforms, this is attributed to the interpolating property of scaling functions. Blurring occurs when insufficient bits are allocated to code fine scale coefficients and the reconstructed signal becomes essentially an interpolated version of a coarser signal. This results in an increase in the transition width of an edge profile, thus affecting the reconstructed edge fidelity.

Ringing, also known as *overshoot* and *peaking*, is another kind of artifact that occurs around discontinuities that are simply wavelet profiles. From quantizing the wavelet reconstruction equation, we can clearly see how the distortion appears as in the following:

$$\hat{f} = \sum_k \langle f, \phi_{0,k} \rangle + Q(\sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}) \tag{5.12}$$

where Q is some quantization function. Rewriting,

$$\begin{aligned}
\hat{f} &= \sum_k \langle f, \phi_{0,k} \rangle + \sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k} \\
&\quad - \left\{ \sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k} + Q(\sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}) \right\} \\
&= \sum_k \langle f, \phi_{0,k} \rangle + \sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k} - (1 - Q) \sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k} \\
&= f - (1 - Q) \sum_j \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}
\end{aligned}$$

Thus the nature and extent of ringing will depend on the length and profile of the wavelet functions weighted by $(1 - Q)$. From an image point of view, such overshoots are manifested as ringing artifacts around the point of discontinuity. It appears around edges because these features contains much high frequency components. It would appear that one can use shorter wavelets to reduce the extend of blurring and ringing. Such choices would limit us to use wavelets with smaller number of vanishing moments which contradicts the choice of wavelets desired for sparse representation in signal

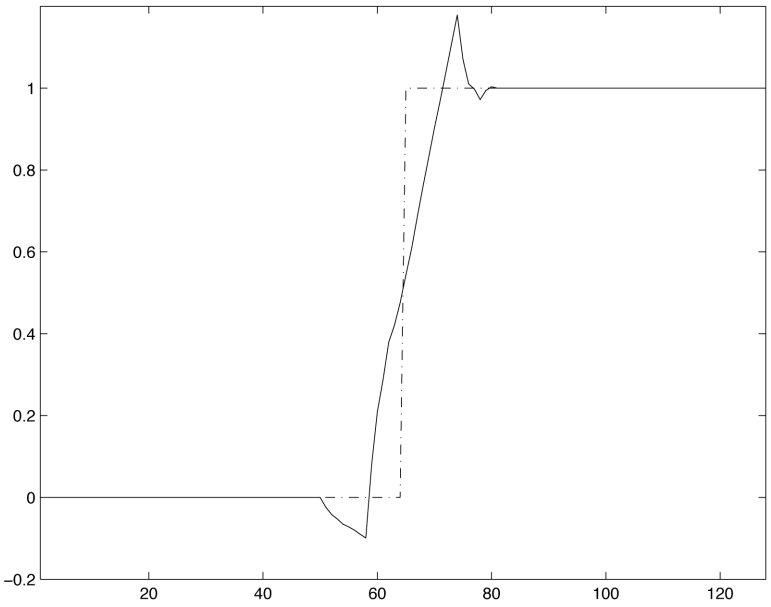


Figure 5.3: Step function reconstructed only from its scaling coefficients. Note the distortion around the step edge at $u = 64$. The step transition width has increased, causing smoothing effect. Overshoots are also observed around the step.

coding.

5.4.2 Wavelet Footprints

Wavelet footprints exploits the concept behind the cone of influence of wavelet coefficients. Dragotti *et al.* [94] proposed that wavelet coefficients in the same cone of influence should be grouped and jointly coded as a vector. These vectors of wavelet coefficients are known as *wavelet footprints*. Suppose J levels of wavelet decomposition with filter of length L, a footprint f_k at location k will be a $(J) \times (L-1)$ vector, containing $J(L-1)$ wavelet coefficients in the neighborhood of k. A wavelet footprint is defined [95] as follows:

Definition 5.4.1. *Given a piecewise constant signal x with only one discontinuity at position k, we call footprint $f_k^{(0)}$ the scale-space vector obtained by gathering together all the wavelet coefficients $\{d_{j,k}\}$ in the cone of influence of k and then imposing $\|f_k^{(0)}\| = 1$. Expressed in the wavelet basis, this footprint can be written as $f_k^{(0)}[n] = \sum_{j=1}^{J} y_{j,k_j} \psi_{j,k_j}[n]$, where $y_{j,k_j} = d_{j,k_j} / \sqrt{\sum_{j=1}^{J} d_{j,k_j}^2}$.*

If a signal consists of polynomials of maximum degree D, then there will be D possible

footprints $f_k^{(d)}$ for $d = 0, 1, 2, \ldots, D$ at each location k. Reconstructing the signal would only require some linear combination of the footprints:

$$\tilde{f}(n) = \sum_k c_{J,k}\phi_{J,k}(n) + \sum_k \sum_{d=0}^{D} \alpha_k^{(d)} f_k^{(d)} \tag{5.13}$$

where $\alpha_k^{(d)} = \langle f, f_k^{(d)} \rangle$ are the footprint coefficients. To reconstruct the signal, it is sufficient to code the footprint positions k and the coefficients $\alpha_k^{(d)}$. The advantage of wavelet footprints is the reduction of coefficients to be coded— a single footprint coefficient and a positional information replaces the need to code an entire cone of wavelet coefficients. However, this technique is not free from wavelet artifacts. If the footprint coefficients are scalar-quantized independently of the scaling coefficients $c_{J,k}$, ringings will be present in the neighborhood of discontinuities. In situations where the footprints' cone of influence overlap, the sub-footprints $\tilde{f}_k^{(d)}$, i.e., only the first $J_1 < J$ row elements of $f_k^{(d)}$, are used, and the error $\sum_{d=0}^{D} f_k^{(d)} - \tilde{f}_k^{(d)}$ can give rise to ringing artifacts.

5.4.3 ENO Wavelets

Developed by Osher *et al.* [96], the *Essentially Non-Oscillatory* (ENO) scheme was originally used in fluid dynamics applications to capture shocks, rarefaction, and contact discontinuities accurately. Their application to data compression was studied by Amat *et al.* [97]. The ENO wavelet transform was introduced in [98][99] as a technique to yield distortion-free edge reconstruction by not filtering across discontinuities. Instead the signal is polynomial-extrapolated about the discontinuities and the filters are applied across the smoother region of the signal at different decomposition scales. As a result, smaller wavelet coefficients are expected. The degree of extrapolation depends on the accuracy order of the wavelets [100].If we use wavelets with p vanishing moments for analysis, a p^{th} order polynomial extension (see section 4.1.3) can be applied.

Unlike wavelet footprints, the objective of the ENO wavelet transform is to generate small or null wavelet coefficients in the vicinity of discontinuities. This implies that there will be little or no distortion in the reconstructed signal if these coefficients are discarded or severely quantized. ENO wavelets require a minimal distance between

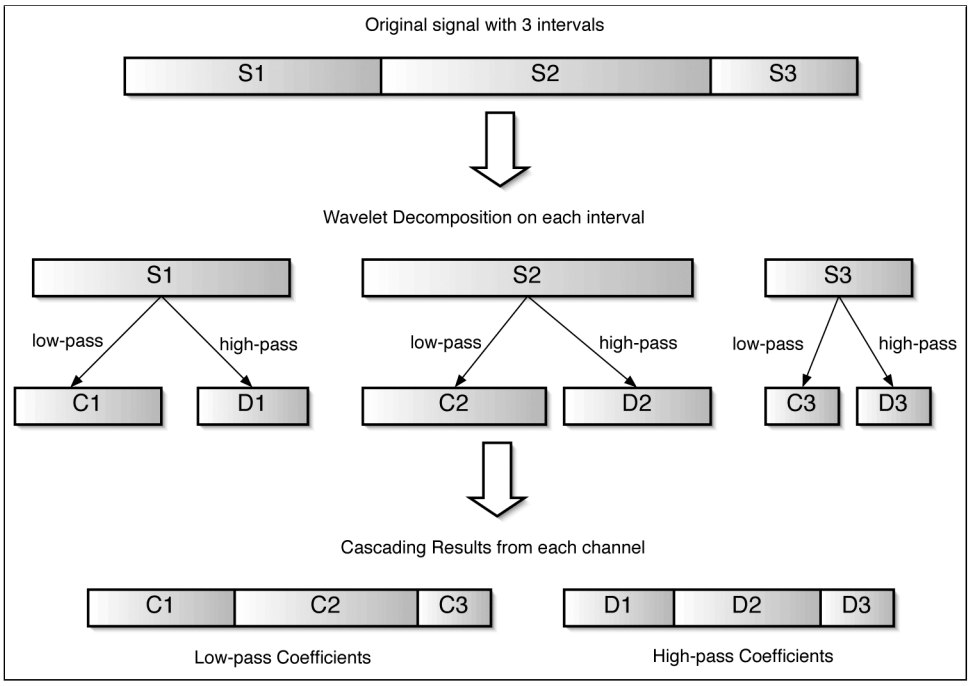


Figure 5.4: Interval wavelet decomposition example.

consecutive discontinuities for distortion-free edge approximation such that the filter does not overlap two extrapolated borders simultaneously. Otherwise, the ENO wavelet would have similar performance as the standard wavelet transform.

Due to the two extrapolated sections for each discontinuity, it seems that the ENO wavelet transform is expansive— the number of output coefficients is greater than the input signal. In section 4.1.3, we have seen that for direct extrapolation, unless we kept all the redundancies in transform, it is impossible to have perfect reconstruction. To overcome this limitation, Chan *et al.* [98][99] proposed extrapolating coefficients at the next coarser scale, which implicitly determines the extrapolating values at the finer scale. Since the ENO wavelets technique is based heavily on the idea of polynomial extrapolation, we can simplify the ENO scheme using the wavelet extrapolation [81][101] to avoid extrapolation at every decomposition level.

5.4.4 Interval Wavelets

The interval wavelets discussed in section 5.4.4 are originally designed for analyzing finite signals. Here, we propose utilizing them to yield a sparser representation for piece-wise smooth signals. The idea is to break up the signal into smaller intervals and

perform independent analysis on each of the segments (see figure 5.4). Consider a finite signal $f \in \mathbb{L}^2[0, N]$ and assuming M break points such that $0 < k_1 < k_2 < \ldots < k_M < N$, then for each intervals

$$f_{k_i} = \mathbf{1}_{[k_{i-1}, k_i - 1]} f$$

such that

$$f = \sum_{i=1}^{M+1} f_{k_i}, \quad k_0 = 0 \text{ and } k_{M+1} = N + 1.$$

The interval wavelets coefficients is given by

$$d_j[n] = W^{\text{int}} f(2^j, n) = \sum_{i=1}^{M+1} W^{\text{int}} f_{k_i}(2^j, n) = \sum_{i=1}^{M+1} \langle f_{k_i}, \psi^{\text{int}}_{j,n} \rangle, \tag{5.14}$$

where ψ^{int} is defined in eqn. (4.19). Equivalently, the interval scaling coefficients are denoted by

$$a_j[n] = \sum_{i=1}^{M+1} \langle f_{k_i}, \phi^{\text{int}}_{j,n} \rangle. \tag{5.15}$$

The follow two-scale equations outline the detail calculations to be perform for each interval f_{k_i} recursively.

Theorem 5.4.1. *For Type-I boundary wavelet (p vanishing moments) filters* $h^{int} = \{H^{left}, h^{left}, h, h^{right}, H^{right}\}$ *(low-pass) and* $g^{int} = \{G^{left}, g^{left}, g, g^{right}, G^{right}\}$ *(high-pass), the two-scale relations are given by:*

For the left boundary $0 \leq k < p$,

$$a_j[k] = \sum_{l=0}^{p-1} H^{left}_{k,l} a_{j-1}[l] + \sum_{m=p}^{p+2k} h^{left}_{k,m} a_{j-1}[m], \tag{5.16}$$

$$d_j[k] = \sum_{l=0}^{p-1} G^{left}_{k,l} a_{j-1}[l] + \sum_{m=p}^{p+2k} g^{left}_{k,m} a_{j-1}[m], \tag{5.17}$$

For internal of the signal $p \leq k < 2^{-j}N - p$,

$$a_j[k] = \sum_{l=-\infty}^{+\infty} h[l-2k]a_{j-1}[l], \tag{5.18}$$

$$d_j[k] = \sum_{l=-\infty}^{+\infty} h[l-2k]d_{j-1}[l]. \tag{5.19}$$

For the right boundary $-p \leq k < 0$,

$$a_j[2^{-j}N+k] = \sum_{l=-p}^{-1} H_{k,l}^{right} a_{j-1}[2^{-j+1}N+l] \tag{5.20}$$

$$+ \sum_{m=-p+2k+1}^{-p-1} h_{k,l}^{right} a_{j-1}[2^{-j+1}N+m], \tag{5.21}$$

$$d_j[2^{-j}N+k] = \sum_{l=-p}^{-1} G_{k,l}^{right} a_{j-1}[2^{-j+1}N+l] \tag{5.22}$$

$$+ \sum_{m=-p+2k+1}^{-p-1} g_{k,l}^{right} a_{j-1}[2^{-j+1}N+m]. \tag{5.23}$$

If it is chosen such that the break points k_i are at the desired discontinuities, then like ENO wavelets, filters are not applied across these discontinuities. Hence, it is possible to generate little or no wavelets coefficients in the vicinity of the signal 'jumps' using interval wavelet decomposition at appropriate breakpoints. To determine the break points, singularity or edge detection is a pre-requisite step. This is also required for both ENO wavelets and wavelet footprints. Various methods are possible such as Laplacian operator and modulus wavelet maximas (see section 5.3.1). In the case of singularity detection using wavelets methods, it is be possible to incorporate both the detection and decomposition in a single stage for application purposes.

The interval wavelet reconstruction is simply obtained from the adjoint equations found in theorem 5.4.1. Each individual segments are reconstructed by their respective

scaling and wavelet coefficients. The approximated signal is obtained by simply cascading and post-filtering these segments. The following theorem gives equations for fast dyadic transform with interval wavelets for a finite signal, with discontinuities assumed at its borders.

Theorem 5.4.2. *The adjoint equations of theorem 5.4.1 is given by;*

For $0 \le l \le p-1$,

$$a_{j-1}[l] = \sum_{k=0}^{p-1} H_{k,l}^{left} a_j[k] + \sum_{k=0}^{p-1} G_{k,l}^{left} d_j[k].$$

For $p \le l \le 3p-2$,

$$\begin{aligned} a_{j-1}[l] &= \sum_{k=\frac{1}{2}(l-p)}^{p-1} h_{k,l}^{left} a_j[k] + \sum_{k=-\infty}^{+\infty} h[l-2k] a_j[k] \\ &\quad + \sum_{k=\frac{1}{2}(l-p)}^{p-1} g_{k,l}^{left} a_j[k] + \sum_{k=-\infty}^{+\infty} g[l-2k] d_j[k]. \end{aligned}$$

For $3p-1 \le l \le 2^{-j+1}N - 3p$,

$$a_{j-1}[l] = \sum_{-\infty}^{+\infty} h[l-2k] a_j[k] + \sum_{-\infty}^{+\infty} g[l-2k] d_j[k].$$

For $-p-1 \ge l \ge -3p+1$,

$$\begin{aligned} a_{j-1}[2^{-j+1}N + l] &= \sum_{k=-p}^{\frac{1}{2}(l+p-1)} h_{k,l}^{right} a_j[2^{-j}N + k] + \sum_{-\infty}^{+\infty} h[l-2k] a_j[2^{-j}N + k] \\ &\quad + \sum_{k=-p}^{\frac{1}{2}(l+p-1)} g_{k,l}^{right} d_j[2^{-j}N + k] + \sum_{-\infty}^{+\infty} g[l-2k] d_j[2^{-j}N + k]. \end{aligned}$$

For $-1 \ge l \ge -p$,

$$a_{j-1}[2^{-j+1}N + l] = \sum_{k=-p}^{-1} H_{k,l}^{right} a_j[2^{-j}N + k] + \sum_{k=-p}^{-1} G_{k,l}^{right} d_j[2^{-j}N + k].$$

Compared to ENO wavelet transform, the advantage of interval wavelet transform does not require any explicit extrapolation or modification to the decomposition steps.

Calculations of the wavelet coefficients is exactly the same as the fast dyadic transform with no increase in computational complexity. One only needs to switch to boundary filters when computing coefficients near the discontinuities.

Now we will give an approximation error bound for interval wavelets decomposition. Given a function $f \in \mathbf{L}^2$, standard wavelet theory shows that it can be linearly approximated by its projection f_j in V_j. This linear approximation has a standard error estimate given by the following theorem.

Theorem 5.4.3. *Suppose the wavelet $\psi(x)$ generated by scaling function $\phi(x)$ has p vanishing moments, and $f_j(x)$ is the approximation of $f(x)$ in V_j with basis $\psi_{j,k}(x)$, then,*

$$\|f(x) - f_j(x)\| \leq C(\delta x)^p \|f^{(p)}(x)\|, \tag{5.24}$$

where $\delta x = 2^{-j}$ and C is a constant independent of j.

The theorem shows that the approximation error is governed by two factors: the spatial resolution δx to the p^{th} power and the norm of the p^{th} derivative of f. For piecewise continuous functions, the error estimate does not give a real upper bound since $\|f^{(p)}(x)\|$ is infinite.

For interval wavelet transform, the approximation coefficients are by default piecewise. Thus the error estimate could be obtained without taking derivatives across discontinuities. Denoting a $(p-1)^{\text{th}}$ order polynomial $q_{p-1}(x)$ which represents the first p terms of the Taylor expansion of $f(x)$ at the origin:

$$f(x) = q_{p-1}(x) + \frac{f^{(p)}(\delta x)}{p} x^p, \tag{5.25}$$

where $f^{(p)}$ is the p^{th} derivative of f. The approximation function is given by

$$\hat{f_j}(x) = \sum_n c_{j,n} \phi_{j,n}(x). \tag{5.26}$$

From the construction of interval wavelets, we know that $\phi_{j,k}^{\text{left}}$,$\phi_{j,k}$ and $\phi_{j,k}^{\text{right}}$ together, with p vanishing moments, can generate all polynomials up to degree $p-1$ on $[0,1]$.

Hence, the approximation error for each interval can be obtained:

$$\begin{aligned} |f(x_0) - \hat{f}_j(x_0)| &\leq |f(x_0) - q_{p-1}(x_0)| + |q_{p-1}(x_0) - \hat{f}_j(x_0)| \\ &\leq C(\delta x)^p \|f^{(p)}(x)\| + |q_{p-1}(x_0) - \hat{f}_j(x_0)| \\ &= C(\delta x)^p \|f^{(p)}(x)\|, \end{aligned}$$

which is identical to that of the continuous smooth signal on the whole-line. Since interval wavelet decomposition treats a piecewise smooth signal as a union of intervals, the approximation error can be written as:

$$\|f(x) - f_j(x)\| \leq C(2^{-j})^p \|f^{(p)}(x)\|_{(0,1)\setminus D}, \tag{5.27}$$

where D is a set of interval breaks in $[0, 1]$.

Thus this proof demonstrates that we can have a error estimate bound of a piecewise continuous function to be the same as a continuous function as if the discontinuities are non-existing. From approximation perspective, this would be the best possible error bound for discontinuous functions.

5.4.5 Discontinuities in Proximity

A major drawback for the techniques described previously, i.e., wavelet footprints, ENO wavelets and the interval wavelets, is that there is a minimum permissible distance between discontinuities for decomposition. This constraint limits the number of decomposition allowed for signal transform. In signal compression, it is favorable that we could apply as many decomposition levels as possible to a signal to obtain compact representations.

The wavelet footprint dictionary contains footprints from a singular cone of influence. Discontinuities that are close will have overlapping cone of influence and hence, their footprints are not represented by any entries in the dictionary. A solution is to represent subsets, in terms of scale, of wavelet footprints of these neighboring discontinuities (see section 5.4.2).

For ENO wavelets, the *discontinuity separation* (DS) property [98] has to be satisfied,

which imposes a constraint on the minimum distance between any two discontinuous points such that a wavelet filter will not overlap them simultaneously. In regions where the DS property is not satisfied, standard wavelet transform is applied, neglecting the presence of the discontinuities. Thus at the places where the DS property is invalid, the the approximations produced by the ENO wavelet transforms are comparable to that by the standard wavelet transforms.

For interval wavelets, there are several ways to overcome this limitation. First, one can adopt the similar approach for ENO wavelets by ignoring the singularities that are too close, merging the small intervals with their neighboring intervals. Alternatively, since each interval is filtered independently, one can adopt shorter boundary filters. With the existing constructed families of interval wavelets, even for Daubechies $p = 2$ boundary filters, the minimum interval length required for non-overlapping is $2(2p + 1) - 2 = 8$. One can proceed with filters constructed from polynomial scaling and wavelet functions as mentioned in section 4.2. If approximation accuracy can be sacrificed for smaller intervals, there is a simpler solution by using the Haar filters, which are trivially defined on intervals. In the extreme case when the interval is only a single coefficient wide, the coefficient can be just copied to the next coarser scale, or the decomposition can be terminated for that interval.

5.4.6 Odd Length Decomposition

In dyadic wavelet decomposition, it is generally assumed that the signal is even length N_{even} and for each decomposition channel, the low and high-pass, will have $N_{even}/2$ coefficients through dyadic sampling. For odd length N_{odd} signal, the decomposition should ideally result in unequal number of coefficients from each channel such that the total number of coefficients is preserved, i.e., $\lfloor N_{odd}/2 \rfloor + \lceil N_{odd}/2 \rceil = N_{odd}$. In practice, the usual treatment is to pad the signal to the next even length prior to decomposition at each scale so that dyadic decomposition could be applied. This introduces some redundancy in the decomposition.

In section 5.4.4, a signal is proposed to be divided into intervals at appropriate breakpoints for decomposition. Thus the interval wavelet decomposition has to be robust in

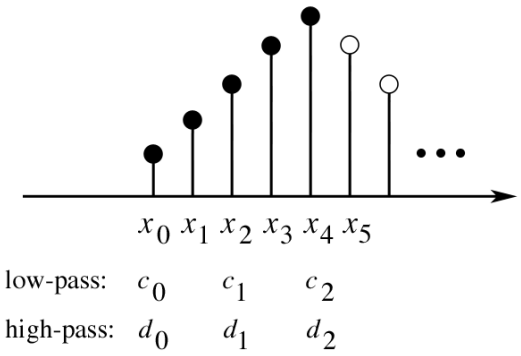


Figure 5.5: Haar filter with odd length sequence $\{x_k\}_{k=0,\ldots,4}$ and whole-point symmetry extension. Black and white circles indicates original and extended sequence respectively.

order to accommodate even and odd lengths of signal. We begin by looking at the case of using Haar filters. We can extrapolate the sequence by whole symmetry. This results in $\lceil N_{odd}/2 \rceil$ non-zero coefficients from each channel, the decomposition expands the number of coefficients since $2\lceil N_{odd}/2 \rceil > N_{odd}$. However, we can abandon one coefficient from either channel to preserve the total coefficient count without compromising perfect reconstruction. In figure 5.5, filtering and downsampling the sequence x leads to 3 low-pass coefficients $\{c_k\}_{0\leq k<3}$ and 3 high-pass coefficients $\{d_k\}_{0\leq k<3}$. For reconstruction, x_4 depends only on either c_2 or d_2 since $x_5 = x_3$. The whole-point symmetry allows $(3, 2)$ (3 low-pass and 2 high-pass coefficients) or $(2, 3)$ coefficients decomposition for the Haar case. Thus only five coefficients are needed for reconstruction, which means that the decomposition remains non-expansive.

For odd-length interval wavelet decomposition with $p \geq 2$, we can utilize the Type-II and III boundary filters introduced in section 4.3. If Type-I filters are applied on the left boundary and Type-II on the right, it will result in one less coefficient from the high-pass channel than the low-pass. Instead, if Type-III filters are used on the right boundary, we will obtain one less coefficient from the low-pass channel than the high-pass. Hence, a combination of Type-I and II or Type-I and III boundary filters will produce a total odd number of coefficients in the interval decomposition. Table 5.1 summarize the type of filters that can be used for odd and even length sequences.

Table 5.1: Application of Boundary Filters for Odd/Even Length Sequences

	Boundary Filter Type		
Sequence Length	Left Border	Right Border	Remarks
even	I	I	$n(c) = n(d)$
even	II	III	$n(c) = n(d)$
even	III	II	$n(c) = n(d)$
odd	I	II	$n(c) = n(d) + 1$
odd	II	I	$n(c) = n(d) + 1$
odd	I	III	$n(c) = n(d) - 1$
odd	III	I	$n(c) = n(d) - 1$

$n(c)$ = number of low-pass coefficients. $n(d)$ = number of high-pass coefficients.

5.5 Numerical Examples

In this section, we give some 1D numerical examples using the interval wavelet transforms. To verify the theoretical error estimate bound in eqn. (5.27), the standard wavelet and interval wavelet decompositions are applied to a piecewise smooth function f with a discontinuity at $x = 512$ to obtain scaling approximates for various levels of decomposition. The order of approximation accuracy is obtained in $\mathbf{L}^2$ and $\mathbf{L}^\infty$ error norm as:

$$O_2(j) = \log_2 \frac{\|f - \hat{f}_j\|_2}{\|f - \hat{f}_{j-1}\|_2} \tag{5.28}$$

and

$$O_\infty(j) = \log_2 \frac{\|f - \hat{f}_j\|_\infty}{\|f - \hat{f}_{j-1}\|_\infty}, \tag{5.29}$$

where the norms are given by

$$\|x\|_2 = \left(\sum_i x_i^2 \right)^{\frac{1}{2}} \quad \text{and} \quad \|x\|_\infty = \max_i |x_i|.$$

Figure 5.6 shows comparisons of the logarithmic plot of the approximation accuracy for both standard and interval wavelet transform . Clearly from the plots, the interval wavelet approximation exhibits accuracy of order 2, 3 and 4 respectively for Daubechies $p = 2, 3$ and 4 wavelets. The standard wavelet approximation, however, do not show

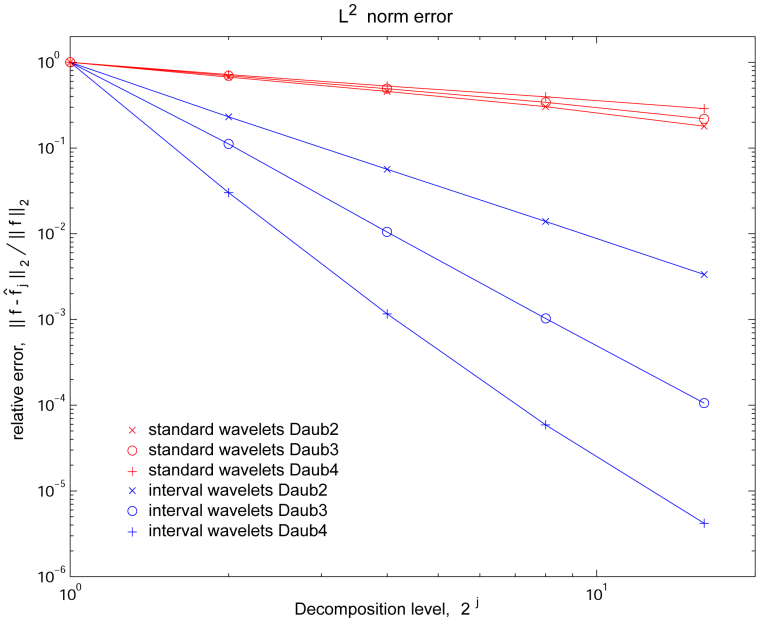
these orders in the presence of the discontinuity in the signal.

Figure 5.7 shows the 1D decomposition and reconstruction example of a piece-wise smooth signal using the standard wavelets, ENO-wavelets and the proposed method. Using interval wavelets, the signal is divided into 9 segments for decomposition where Laplacian operators are used to determine the segment boundaries. Each segment is transformed using Type-I interval wavelets[1]. The signal reconstructed using standard wavelets exhibits the usual ringing artifacts around strong edges. In contrast, these artifacts are absent in the signal approximated using the ENO and the interval wavelet transform. Figure 5.8 shows the coefficients produced by both standard and interval wavelet transform. Note the sparseness of coefficients due to interval wavelet transform.

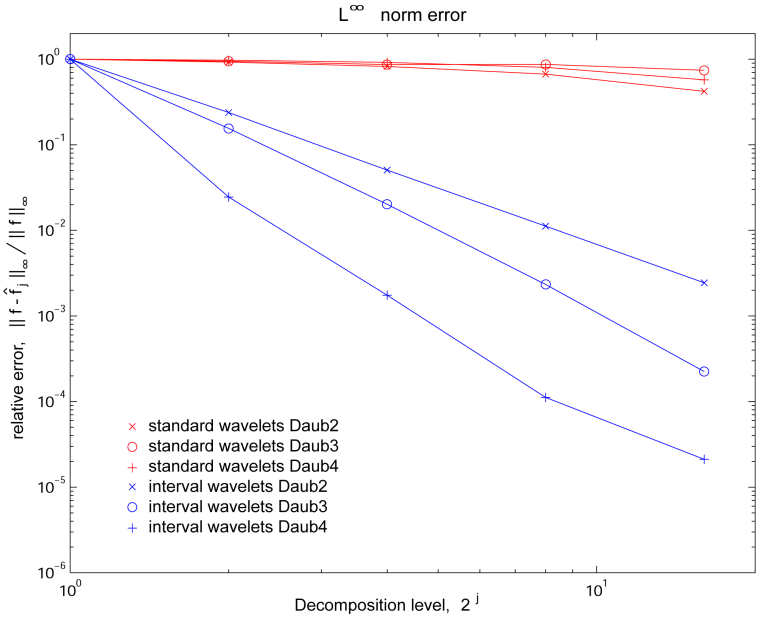
5.6 Summary and Remarks

In this chapter, we have looked at how the smoothness of a signal and its discontinuities affect the wavelet coefficients. The vanishing moments of the wavelets play a major role in suppressing wavelet coefficients of polynomial signals. In the neighborhood of signal discontinuities, the wavelet decay is slower and is related to the uniform and pointwise Lipschitz regularity α of the signal. The decay of the wavelet modulus maxima can be used to measure the local regularity of the signal, hence categorizing the types of singularities within it. It allows us to isolate important discontinuities such as steps and ramps for the purpose of interval wavelet decomposition. Concerning artifacts arising from lossy compression, we have look at existing techniques, both wavelet footprints and ENO wavelets, that are used to minimize the effect. We proposed using interval wavelets to obtain better discontinuity reconstruction by dividing the signal into segments at appropriate location determined by the local regularity measure via wavelet maxima. We show that in the presence of discontinuities, the interval wavelet is able to retain the same approximation accuracy as for globally smooth function. Experiments on non linear signal approximation shows that interval wavelets can reconstruct signals with less artifacts around discontinuities than standard wavelets.

[1]Similar results are obtained for Type-II and III interval wavelets

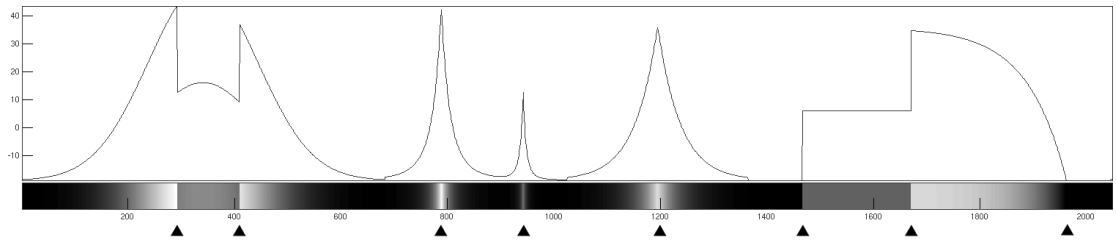


(a)

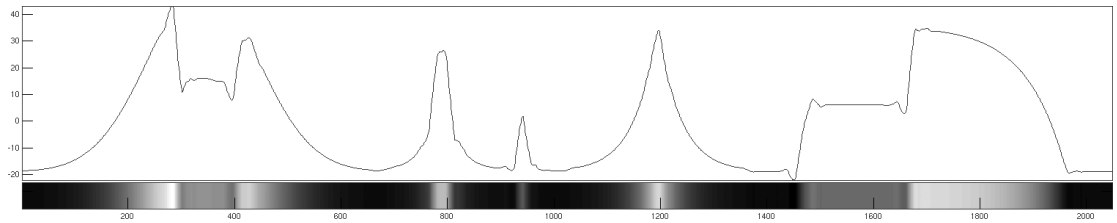


(b)

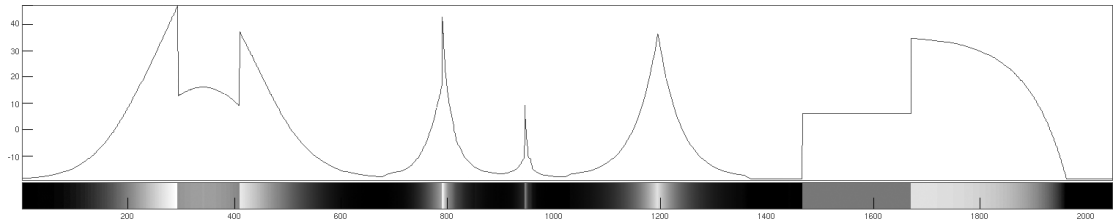
Figure 5.6: Comparison of order of approximation accuracy between standard and interval wavelet transform, on a piecewise smooth function, using (a) $\mathbf{L}^2$ error norm and (b) $\mathbf{L}^\infty$ error norm estimate.



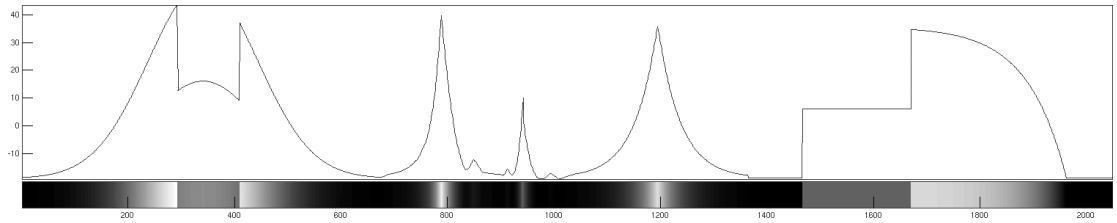
(a) Piecewise regular sequence. (black triangles marked discontinuities positions).



(b) Standard wavelet approximation, MSE = 3.704

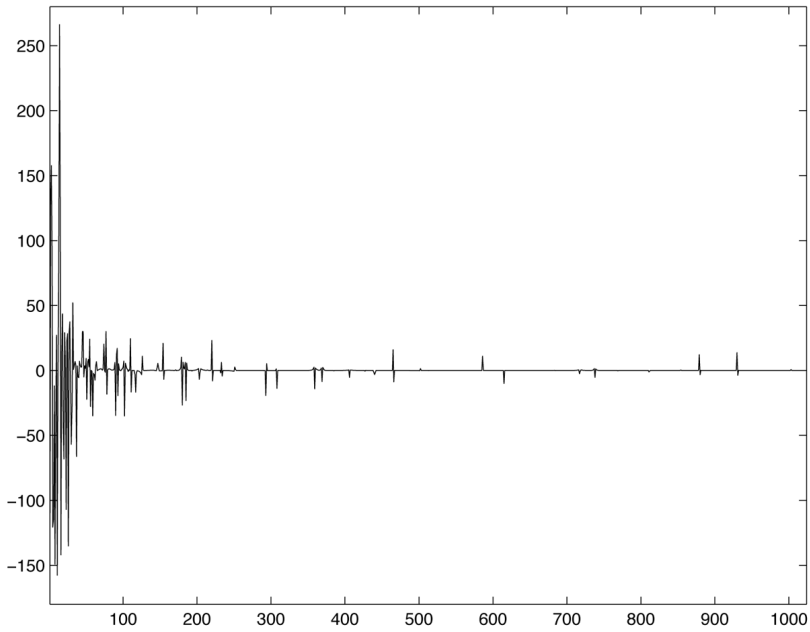


(c) ENO wavelet approximation, MSE = 2.282

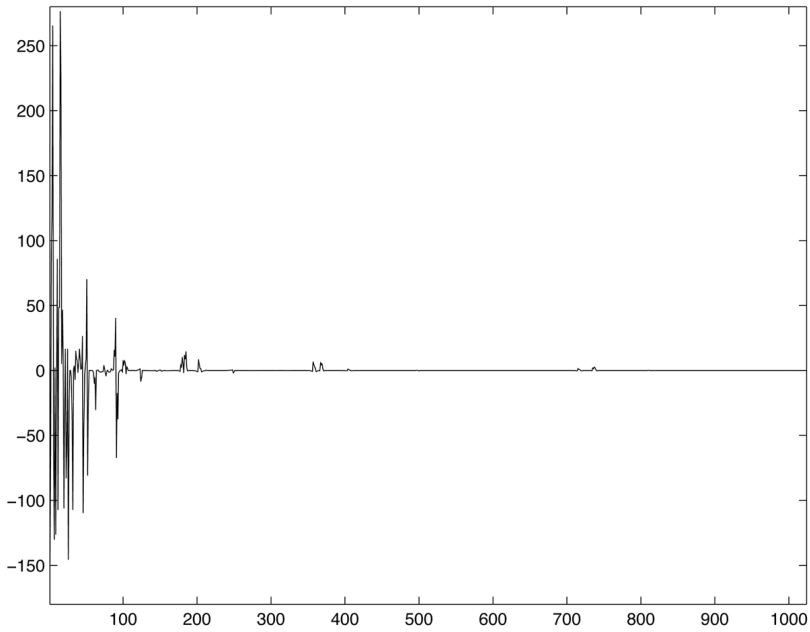


(d) Interval wavelet approximation, MSE = 0.348

Figure 5.7: Signal is approximated using scaling coefficients from 4-level decomposition with Symmlets ($p = 4$) wavelets. Signal discontinuity positions are obtain using second-order laplacian operator.



(a) Standard wavelet coefficients



(b) Interval wavelet coefficients

Figure 5.8: Wavelet coefficients from decomposition of piecewise-regular signal in figure 5.7.

6 Perceptual Image Coding I

Art is the elimination of unnecessary —Pablo Picasso (1881-1939)

In chapter 5, we proposed the application of interval wavelets for analysis and synthesis of 1D piecewise smooth signals. Compared to standard wavelets analysis, interval wavelets give good approximations in the vicinity of singularities without ringing artifacts. Given this observation, it is natural to question the applicability of interval wavelets for image coding. In this chapter, we propose extending interval wavelet analysis to 2D images in order to achieve better approximations for image edges.

In multidimensional data analysis, it is usually straightforward to apply 1D wavelet decomposition to each dimension of the data separately. Despite being less optimal than non-separable wavelet method, this technique is widely used for its simplicity. It is thus desirable to obtain a fast 2D interval wavelet decomposition by applying eqns. (2.51) and (2.52) recursively to each row and column data of an image, in a similar manner for standard 2D wavelet decomposition. However, a direct 2D implementation is not possible without some technical difficulties. We looked at three specific implementation issues:

- *Unbalanced decomposition*. In non-expansive decomposition of odd length $2n + 1$

intervals, either the low or high-pass channel will have one more coefficient than the other. If the low and high-pass channel consistently yield $n+1$ and n coefficients respectively all odd-length intervals in a signal, this can lead to, overall, a very unbalanced decomposition where there are significantly more low-pass coefficients than high-pass coefficients.

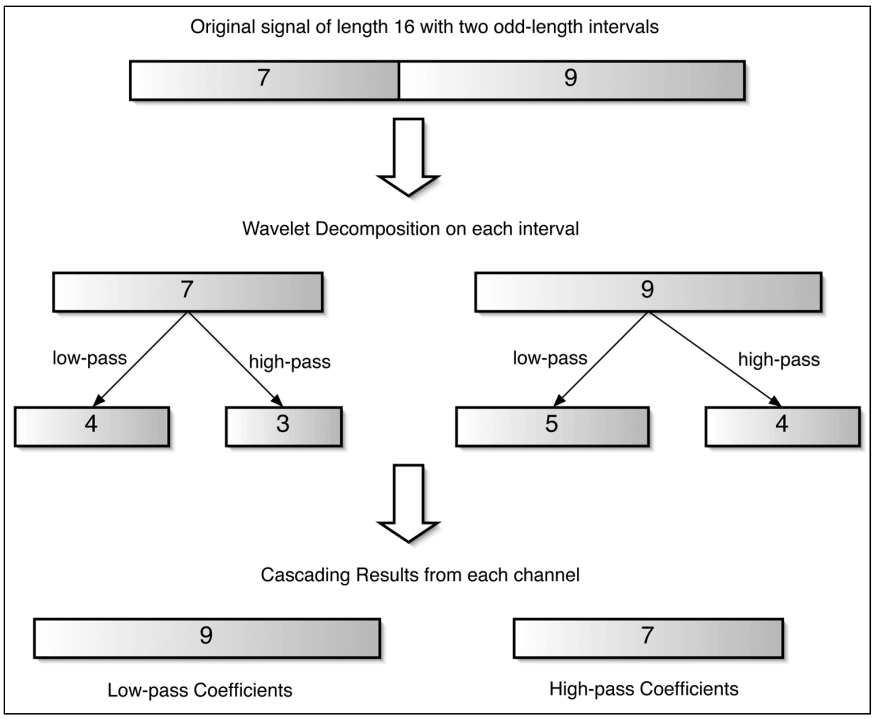
- *Filter misalignment*. Each row or column data will have different number of intervals at various locations. If each of these intervals is transformed independently, the interior filters between each row or column will not be aligned. This can lead to significant visual artifacts in the reconstructed images from quantized coefficients.

- *Non optimal edge locations*. To divide a signal into intervals, edge detection is needed to determine the breakpoints. Our experiments show that direct utilization of these detected edge locations does not produce the desired sparse representations for images.

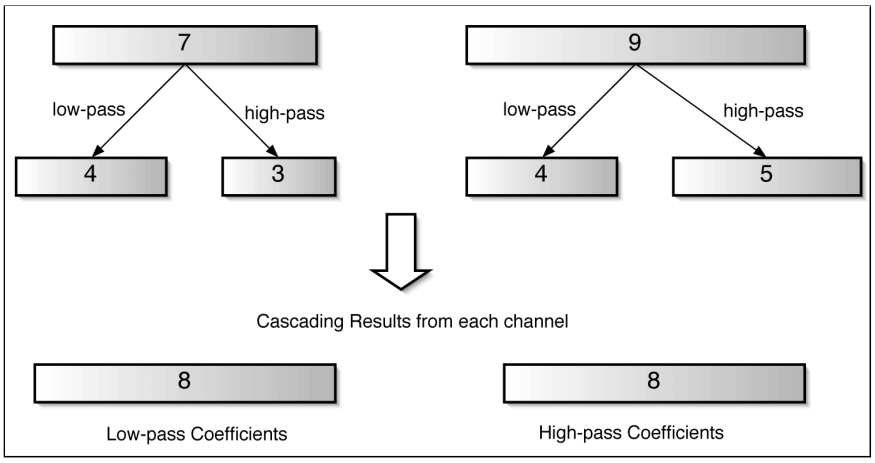
These problems are discussed in detail in the following sections while the 2D interval wavelet decomposition and reconstruction algorithms are discussed in section 6.4.

6.1 Balanced Decomposition

In section 5.4.4, it is proposed that an ensemble of Type I, II and III interval filters can allow a non-expansive interval wavelet decomposition of arbitrary length sequences. This is summarized in table 5.1, which lists the appropriate filters to use for odd or even length intervals, i.e., the application of filter pair types (for left and right boundary) depends on the length parity of each interval. In order to keep the decomposition both non-expansive and *balanced* for each level, we need to determine additionally, which filters to use for various intervals at different positions. Figure 6.1 illustrates the problem of unbalanced decomposition. In figure 6.1(a), an even length signal has a discontinuity located such that the two resulting intervals are odd length. From table 5.1, we can consistently apply Type I and II interval filters to both intervals. However, this would lead to an unbalanced distribution of 9 low-pass and 7 high-pass coefficients. The ideal case is to yield equal number of low and high-pass output coefficients since the original signal length is even. If Type I and II interval filters are applied to the first interval, and



(a) Unbalanced decomposition example.



(b) Balanced decomposition example.

Figure 6.1: Examples of unbalanced and balanced decomposition. (a) Consistently applying the same filter order, Type I and II for left and right boundaries, for odd length intervals. (b) Alternating configuration, where Type I and II filters are applied to left and right boundaries of first interval, and Type III and II filters applied to left and right boundaries of second interval.

Type I and III to the second interval, the output coefficients will be balanced as shown in figure 6.1(b).

To ensure that the number of scaling and wavelet coefficients are balanced in addition to having a non-expansive decomposition, we need to predetermine the appropriate number of scaling and wavelet coefficients for each decomposition for every interval. Suppose a well ordered set of M edge locations $E = \{e_k\}_{0 \le k < M} \in [0, N]$ on a finite sequence $\{c_k\}_{0 < k \le N}$ where $e_0 = 0$ and $e_{M-1} = N$, then the corresponding M edge locations in the next coarser scale is given by:

$$\textit{low-pass channel:} \qquad E_c(n) \longleftarrow \lceil E(n)/2 \rceil, \quad 0 \le n < M. \tag{6.1}$$

$$\textit{high-pass channel:} \qquad E_d(n) \longleftarrow \lfloor E(n)/2 \rfloor, \quad 0 \le n < M. \tag{6.2}$$

Through recursive application, both eqns. (6.1) and (6.2) implicitly define the number of scaling and wavelet coefficients to be computed for each interval at different scales. For example in figure 6.1(b), the original sequence has edges located at $\{0, 7, 16\}$. This yields edge locations in the next coarser scale, $\{0, 4, 8\}$ and $\{0, 3, 8\}$ for the low and high-pass sequences respectively. Thus we expect the first interval to have 4 low- and 3 high-pass coefficients, and the second interval to have 4 low- and 5 high-pass coefficients. This gives a total balance of 8 low and high-pass coefficients and one would expect in the case of standard wavelet decomposition of a length-16 sequence.

In the case where two neighboring edges at e_n and e_{n+1} are such that $e_{n+1} - e_n = 1$, we can have either $\lfloor e_n/2 \rfloor = \lfloor e_{n+1}/2 \rfloor$ or $\lceil e_n/2 \rceil = \lceil e_{n+1}/2 \rceil$ in the next coarser scale. Figure 6.2 illustrates the situation. At scale j, the edges in sequence c_j are located at $k = 2$ and $k = 3$. Using eqn. (6.1) and (6.2), the edge locations in c_{j-1} are $k = 1$ and $k = 2$ while d_{j-1} has duplicated edges at $k = 1$. At the next scale, c_{j-2} has duplicated edges at $k = 1$ and d_{j-2} has edges at $k = 0$ (which is trivial) and at $k = 1$. For these single pixel wide intervals, the coefficient is not filtered but copied across all scales with appropriate scaling factor of $\sqrt{2}$.

For 2D image decomposition, since we are performing row and column-wise filtering, we have two binary edge maps representing horizontal and vertical edges. The binary

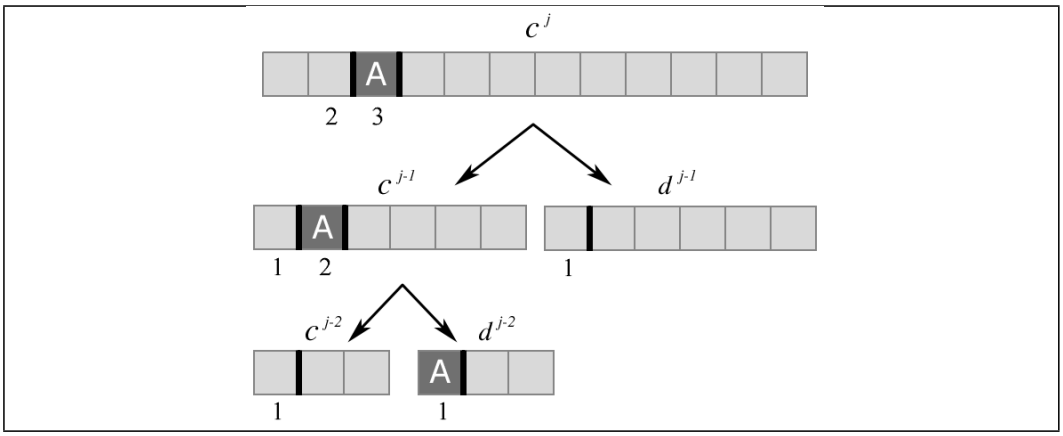


Figure 6.2: For single pixel wide intervals, label 'A', the actual value is not filtered but copied and scale-factored across all scales.

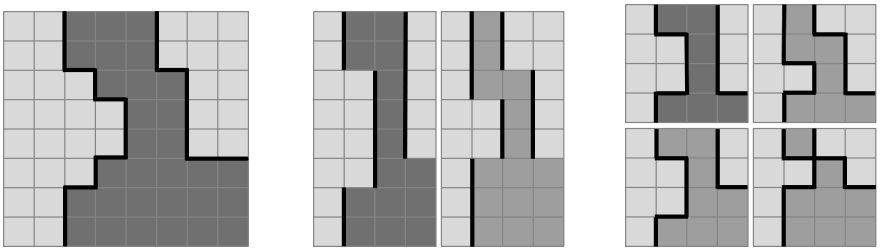


Figure 6.3: (a) Original image, black lines outlines the vertical and horizontal edges; (b) Edge locations after row decomposition. (c) Edge locations after column decomposition.

edge maps of the image have to be encoded so that the decoder is able to determine all interval lengths at different scales using eqns. (6.1) and (6.2) in order to perform signal synthesis correctly. Figure 6.3 illustrates an example of multiscale edge calculation for 2D. In figure 6.3(a), the 3^{rd} row of the image array has an edge located at $k = 3$. After a row-wise decomposition in figure 6.3(b) , the edges are located at $\lceil k/2 \rceil = 2$ and $\lfloor k/2 \rfloor = 1$ in the low-pass and high-pass band respectively. There are various methods to code the binary map, such as arithmetic coding, run-length coding and chain coding. The overheads incurred from coding the binary edge map is not considered in this work.

6.2 Scanline Filter Misalignment

Another problem in a direct 2D implementation of interval wavelet decomposition is that the interior filters h and g may be misaligned with respect to each decomposition rows and columns. Suppose we have two sequences, $\{x_n\}_{0<n\leq N}$ and $\{y_n\}_{0<n\leq N}$ and y has

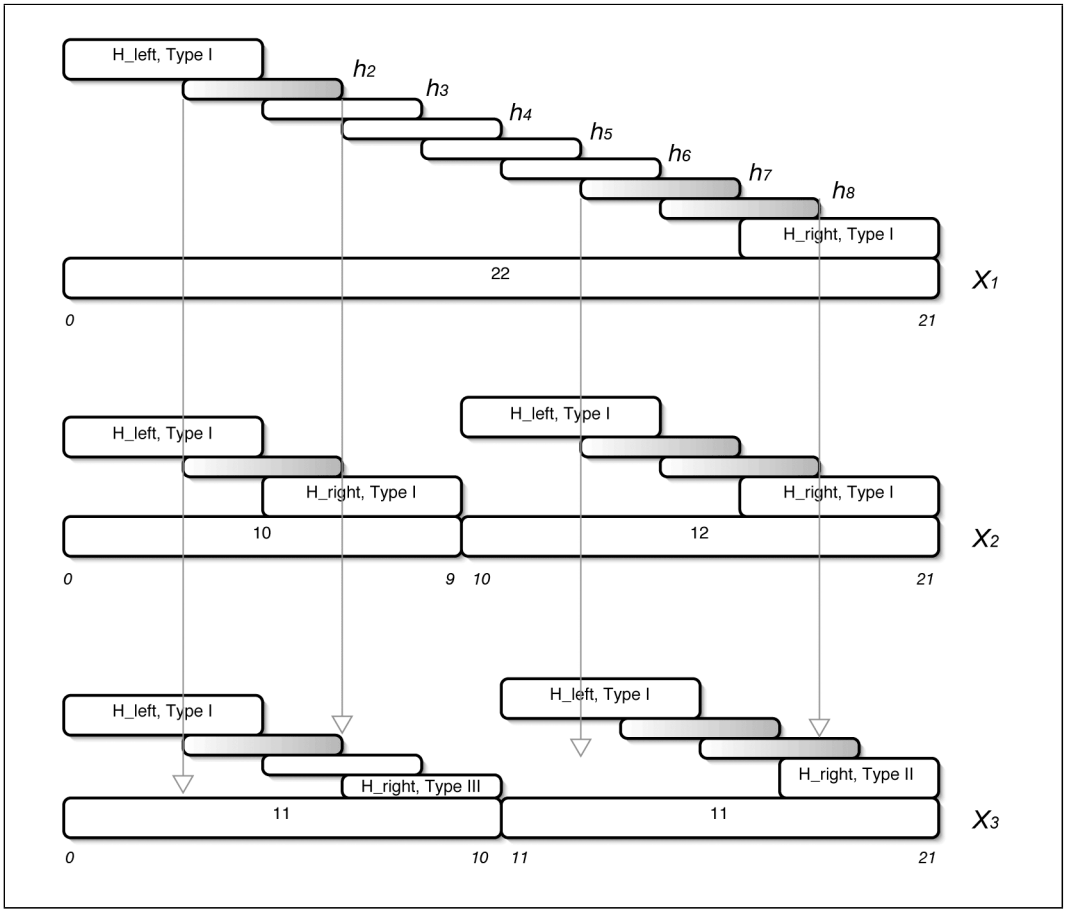


Figure 6.4: Illustrated example for filter misalignment in three consecutive rows of data, X_1, X_2 and X_3. The two even length intervals in X_2 are transformed by Type-I-I interval filters. For both intervals, there is no misalignment of interior filter h (shaded) with respect to those on X_1. In X_3, there are two odd length intervals, analyzed using Type-I-III and Type-I-II filters for balanced decomposition. In the first interval, there is no misalignment in the interior filters h. In the second interval, the filters h are misaligned to those in X_1.

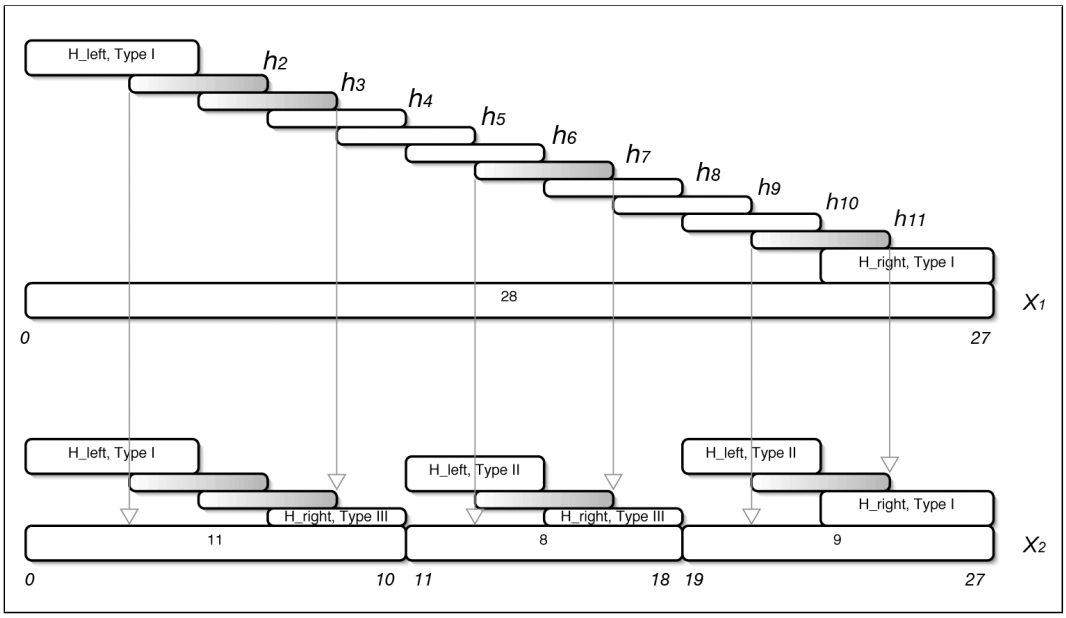


Figure 6.5: Illustrated example for correct filter alignment in two consecutive rows of sequence, X_1 and X_2. X_2 contains three intervals. By applying the appropriate assigned boundary filters (see section 6.2), the interior filters are correctly aligned between X_1 and X_2.

a discontinuity or 'jump' at n_0, i.e., the absolute forward difference $|y_{n_0+1} - y_{n_0}|$ exceeds a certain threshold. Let the two sequences x and y be wavelet transformed using only Type-I ($p = 2$) boundary filters. For sequence x, the scaling coefficients c_n due to interior low-pass filter h are given by:

$$c_n = \sum_{m=0}^{3} h_m x_{2n-2+m}, \quad 2 < n \leq \frac{N}{2} - 2. \tag{6.3}$$

For sequence y, we have two intervals, $\{y_k\}_{0<k\leq n_0}$ and $\{y_k\}_{n_0<k\leq N}$ to be decomposed independently. For even n_0, the scaling coefficients due to interior low-pass filter are expressed similarly as in equation (6.3). For odd n_0, the coefficients from the second interval of sequence y will have a different expression:

$$c_n = \sum_{m=0}^{3} h_m y_{2n-3+m}, \quad \lceil n_0/2 \rceil + 2 < n \leq \frac{N}{2} - 2 \tag{6.4}$$

For the coefficient $c_{\lceil n_0/2 \rceil+2}$ from eqn. (6.4), the low-pass filter h is positioned on $\{y_k\}$ at $k = n_0 + 4$. Compared to coefficient $c_{\lceil n_0/2 \rceil+2}$ from eqn. (6.3), the filter is positioned at $k =$

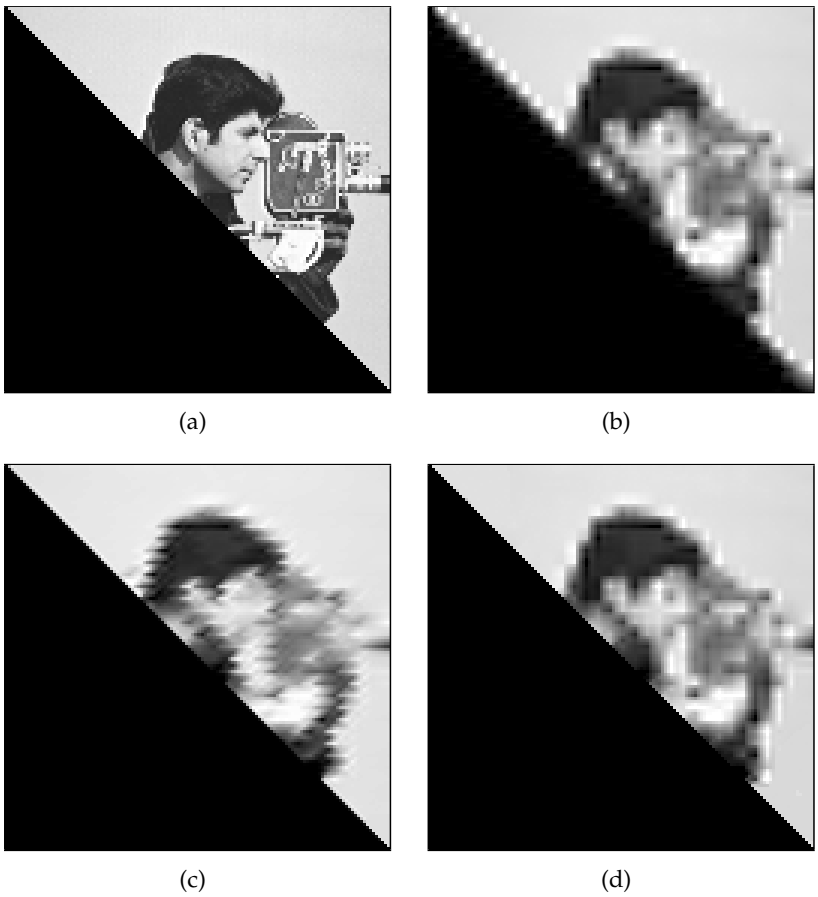


Figure 6.6: (a) Masked *Cameraman* image with diagonal $x = y$ segment boundary profile. Image reconstruction from only scaling coefficients with 2-level (b) standard wavelet decomposition, (c) interval wavelet decomposition without misalignment correction and (d) interval wavelet decomposition with misalignment correction, using Daubechies ($p = 2$) wavelets.

n_0+5 instead. There is an relative offset or *misalignment* between the positions of interior filter h on x and y. Figure 6.4 illustrates the problem. To solve the alignment problem, suppose an interval S from a sequence $\{x_n\}_{0<n\leq N}$ such that $S = \{x_{e_0}, x_{e_0+1}, \ldots, x_{e_1-1}, x_{e_1}\}$ and $0 < e_0 < e_1 \leq N$. Then, assign the left F_1 and F_2 boundary filters as follows:

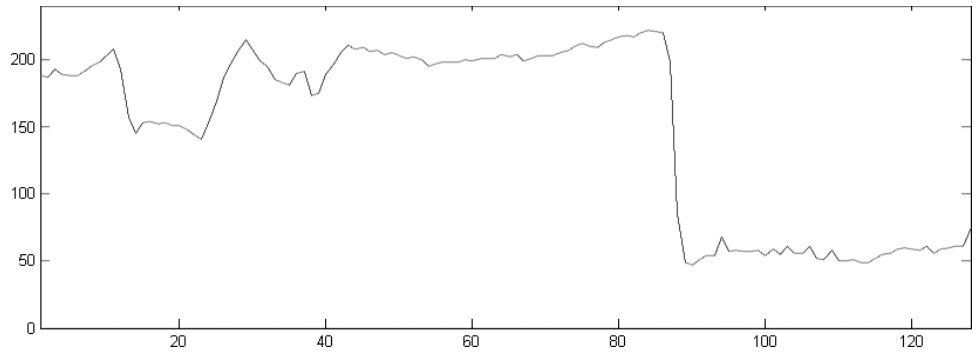
- if e_0 even and e_1 odd, let F_1 =Type-I and F_2 =Type-I.
- if e_0 odd and e_1 odd, let F_1 =Type-II and F_2 =Type-I.
- if e_0 even and e_1 even, let F_1 =Type-I and F_2 =Type-III.
- if e_0 odd and e_1 even, let F_1 =Type-II and F_2 =Type-III.

S can be filtered by F_1 and F_2 boundary filters on the left and right borders respectively. Figure 6.5 shows the correct interior filter alignments. In 2D image decomposition, when interior filters are 'misaligned' between rows or columns, artifacts can be observed in the reconstructed image from quantized coefficients. Figure 6.6 shows a standard and interval wavelet reconstruction of the *Cameraman* with the lower triangular segment masked. Although the image in figure 6.6(c) has very accurate boundary reconstruction, the interior image contains artifacts due to misaligned interior filters. Figure 6.6(d) shows the image reconstruction with correct filter alignment.

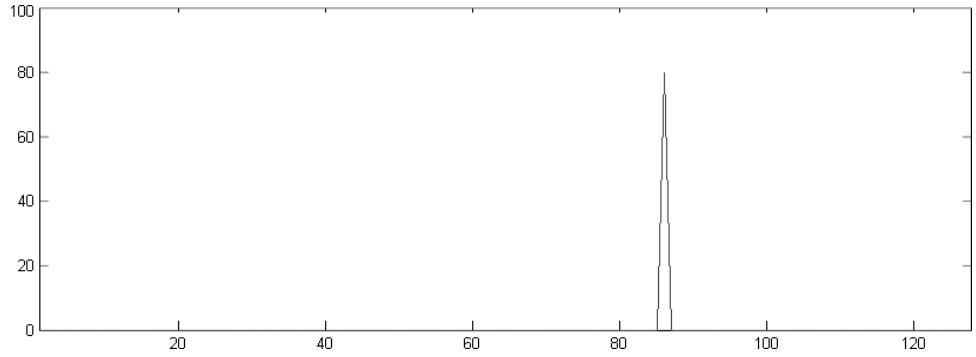
6.3 Edge Jitter Correction

Edge detection is a pre-requisite step that provides information on edge locations for image analysis using interval wavelets. This edge location information is needed to divide each image scanline into intervals. The edge detection can be performed using one of the many well-known techniques such as Canny [102], Sobel, Prewitt, Laplacian operators [103][104][105] and wavelet-based multiscale edge detectors. The multiscale wedgelets and beamlets introduced in chapter 6 can also be used to code edge locations using their parameters for line equations on dyadic partitions.

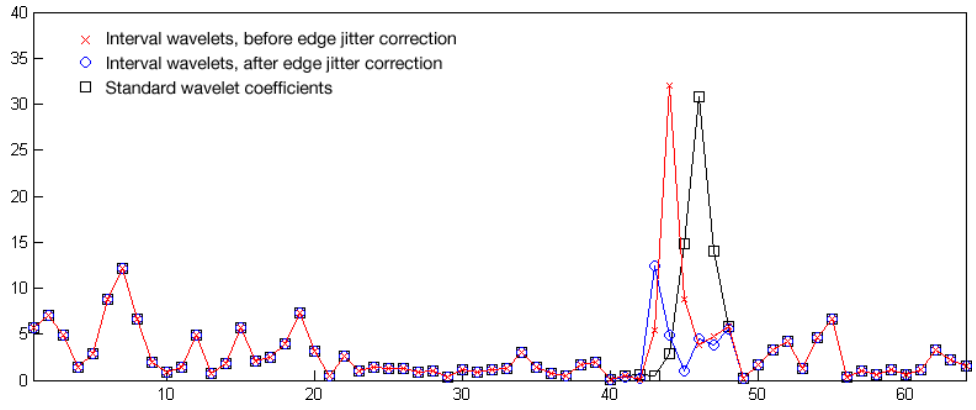
In images, most of the interesting edges have *ramp*-like 1D cross-section profiles than step functions due to the antialiasing effect in digital image sampling and display. Direct utilization of the edge locations obtained from detection techniques may not give optimal decomposition results. An example is illustrated in figure 6.7. The original edge



(a) Original signal obtained from a scanline in *Lena* image.



(b) Absolute wavelet maxima at finest scale.



(c) Comparison of wavelet coefficients.

Figure 6.7: Correction of edge location by jittering. Jitter correction shifts discontinuity from $x = 85$ to $x = 88$. The new edge location leads to significantly smaller coefficients for interval wavelet decomposition. Both standard and interval wavelets decomposition are performed using *Symlets* ($p = 4$).

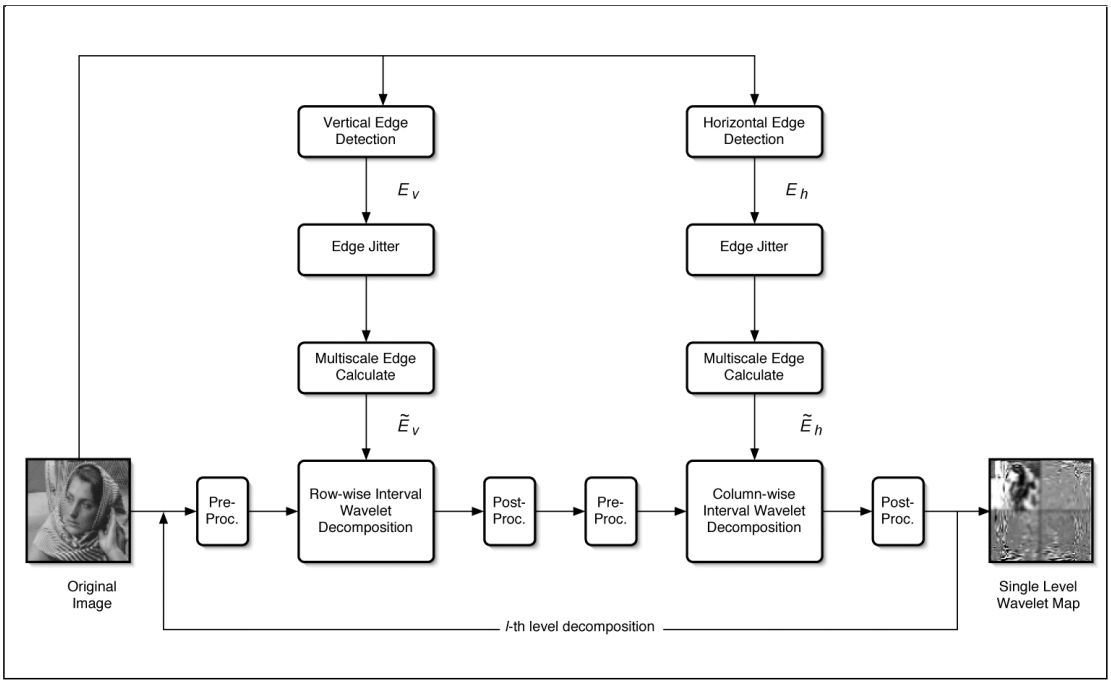


Figure 6.8: Overview of a 2D interval wavelet decomposition.

is detected at $x = 85$, where the signal has a ramp-like profile. Using this default position as a breakpoint for interval wavelet decomposition does not give satisfactory result as the coefficients are comparable to those obtained from standard decomposition.

We propose a 'jittering' strategy to refine the edge position that is adapted to the boundary filters. In other words, a new edge position $\hat{e}$ is obtained within a search window around the detected position e that will minimize the absolute sum of the wavelet coefficients:

$$\min_{e-\Delta\le\hat{e}<e+\Delta}\left(\sum_{m=e-2\Delta}^{e+2\Delta}|\langle\mathbf{1}_{[0,\hat{e}]}f,\psi_{0,m}^{\text{int}}\rangle|^2+\sum_{n=e-2\Delta}^{e+2\Delta}|\langle\mathbf{1}_{[\hat{e}+1,N)}f,\psi_{0,n}^{\text{int}}\rangle|^2\right), \tag{6.5}$$

where f is supported on $[0, N-1]$ and 2Δ is the search window width. Figure 6.7(c) demonstrates the advantage of edge jitter correction with $\Delta = p$, which results in significantly smaller interval wavelet coefficients as compared to both standard wavelet decomposition and interval wavelet decomposition without edge corrections.

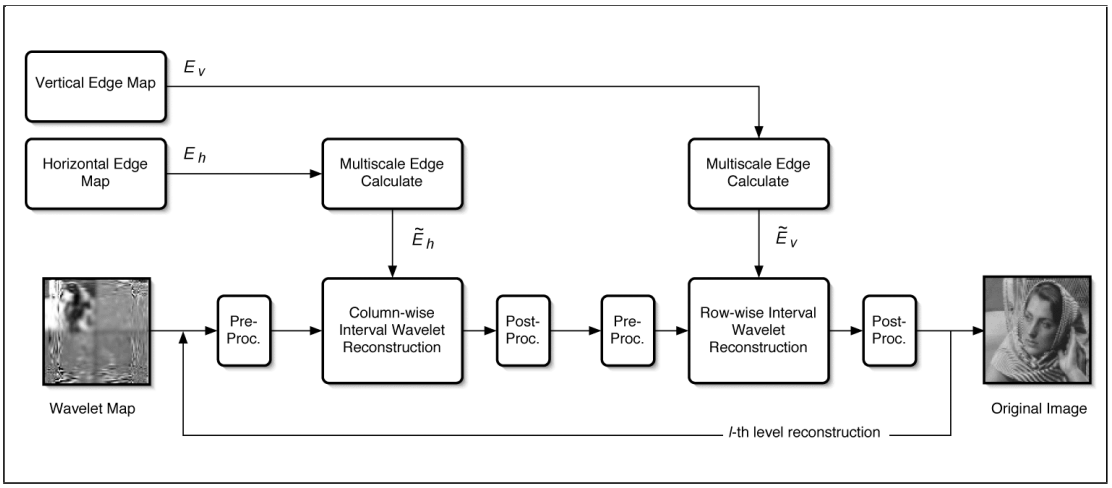


Figure 6.9: Overview of a 2D interval wavelet reconstruction.

6.4 2D Interval Wavelet Decomposition

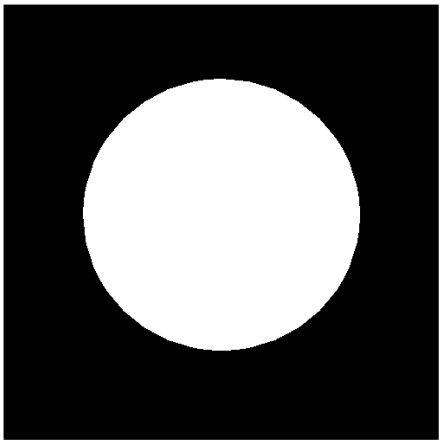
In this section, we utilize the solutions for various problems discussed previously and present procedures for image decomposition using interval wavelets. An overview of the decomposition and reconstruction scheme is given in figure 6.8 and 6.9. Each row and column in the image is broken into segments or intervals, depending on the vertical and horizontal edge boundaries given by binary edge maps E_v and E_h respectively. By exploiting the information from multiscale edge maps $\tilde{E}_v$ and $\tilde{E}_h$, the decomposition is performed on each interval according to the boundary filter assignment guidelines proposed in section 6.2 so that the following properties are fulfilled:

- Non-expansive decomposition.
- Balanced decomposition.
- Aligned interior filters.

In section 4.2.4, we have seen that pre- and post-filtering is needed obtain sparse wavelet coefficients for smooth signals. The pre- and post-filtering steps are performed for each column and row data prior-to and after decomposition at each scale.

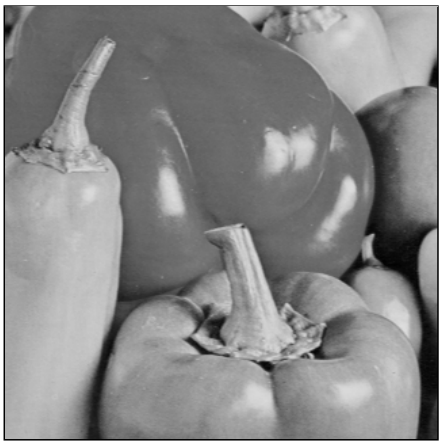
6.5 Numerical Examples

Using test images shown in figure 6.10, we compare the non-linear approximations of images using classical wavelet, wavelet maxima [92] and interval wavelet transform.



(a) *Circle*

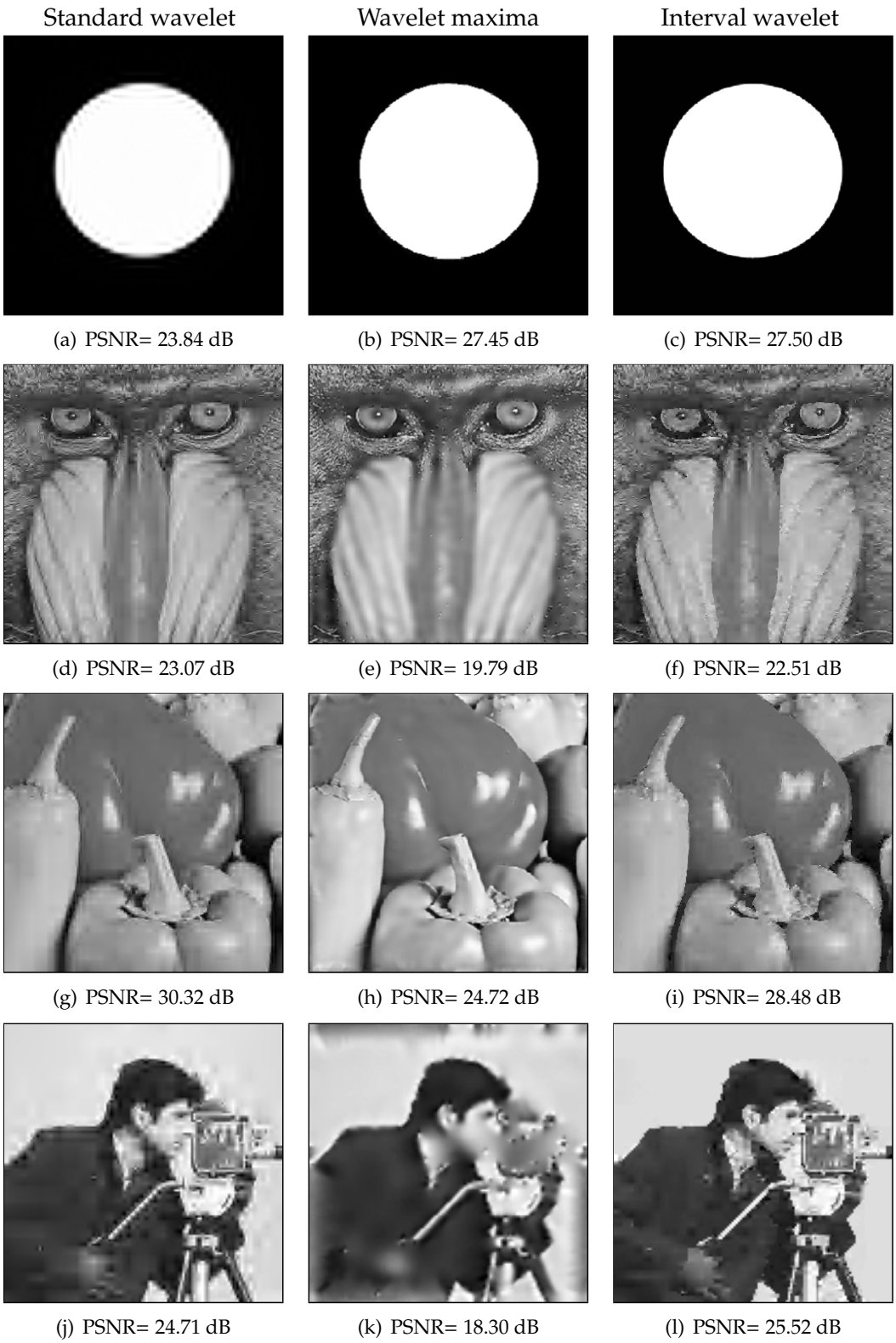
(b) *Mandrill*



(c) *Peppers*

(d) *Cameraman*

Figure 6.10: Original test images



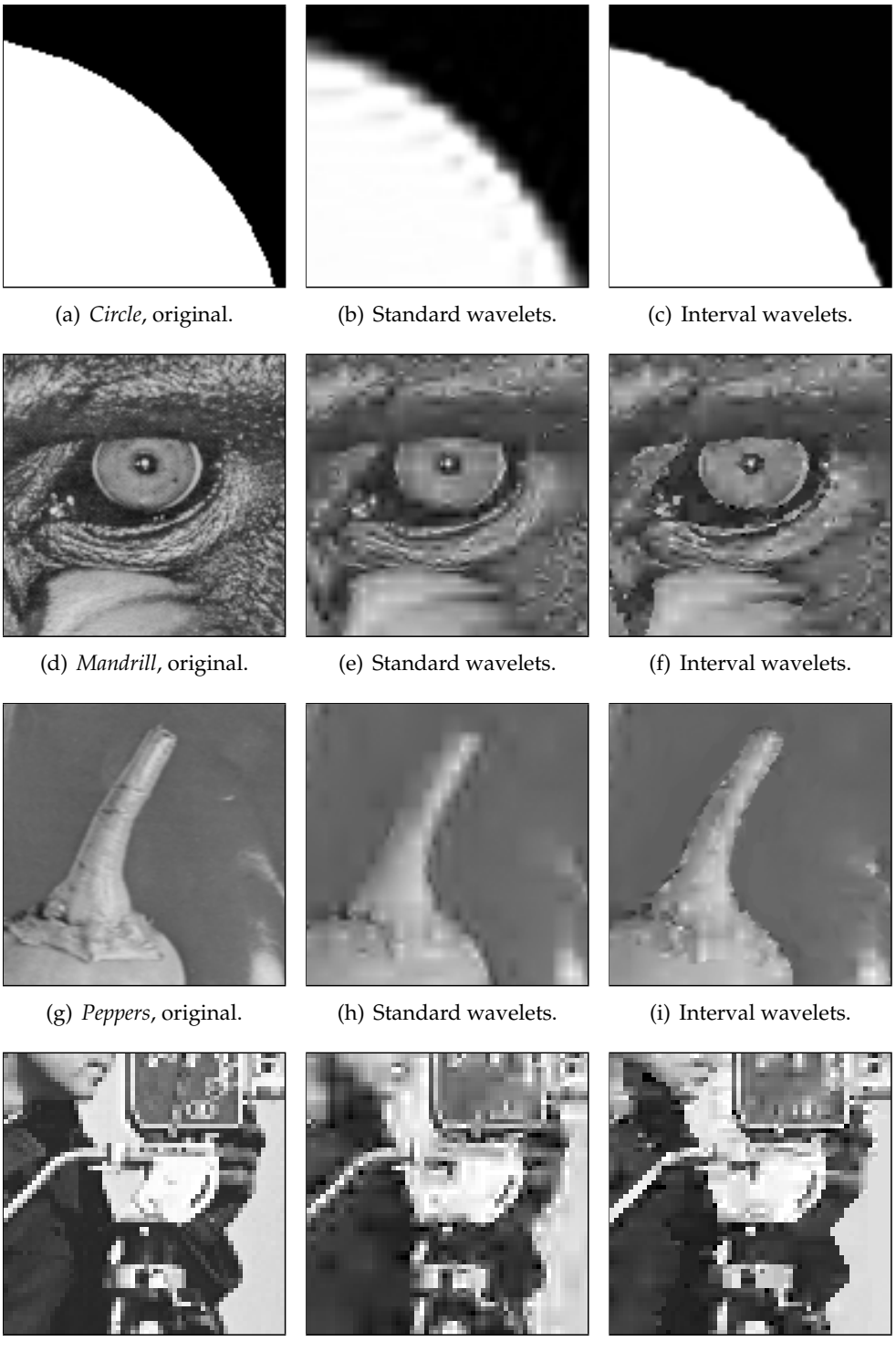
(a) PSNR= 23.84 dB (b) PSNR= 27.45 dB (c) PSNR= 27.50 dB

(d) PSNR= 23.07 dB (e) PSNR= 19.79 dB (f) PSNR= 22.51 dB

(g) PSNR= 30.32 dB (h) PSNR= 24.72 dB (i) PSNR= 28.48 dB

(j) PSNR= 24.71 dB (k) PSNR= 18.30 dB (l) PSNR= 25.52 dB

Figure 6.11: Nonlinear approximations, using 6.25% coefficients, of various images, (a)–(c): Circle, (d)–(f): Mandrill, (g)–(i): Peppers, and (j)–(l): Cameraman, using 3-level decomposition with Daubiechies ($p = 2$) wavelets.



(a) *Circle*, original. (b) Standard wavelets. (c) Interval wavelets.

(d) *Mandrill*, original. (e) Standard wavelets. (f) Interval wavelets.

(g) *Peppers*, original. (h) Standard wavelets. (i) Interval wavelets.

(j) *Cameraman*, original. (k) Standard wavelets. (l) Interval wavelets.

Figure 6.12: Zoomed and cropped versions of the original test and reconstructed images.

The results are shown in figure 6.11 with the close-up of these images given in figure 6.12. The algorithms 6.1 and 6.2 are used for interval wavelet transform. The Sobel edge detector is used to obtain the edge information from *Circle*, *Peppers* and *Cameraman*. A more complicated multi-scale edge detection based on wavelet maxima of quadratic spline wavelets is used for *Mandrill*, in which the edges are detected based on their Lipschitz regularity measure and the propagation depth of the maximas. As expected, approximation by classical wavelets blurs the edges and introduces ringing artifacts along edges. Both the wavelet maxima and interval wavelet transform are able to produce sharper image edges with less artifacts. However, the PSNR values are not consistent with visual observation of the reconstructed images due to limitation of PSNR measures.

For coarser approximations, it is observed that interval wavelets tend to produce images that have sharper edges than the original images. This would lead to images that seem segmented or *'cartoonized'*. Therefore, an original anti-aliased image edge will appear aliased or *'jagged'* as in figure 6.11(l). The jagged edges can be remedied by well-known antialiasing procedures [106][107][108] found in computer graphics literature since the edge locations are known.

6.6 Summary and Remarks

In this chapter, various issues encountered in the extension of interval wavelet analysis from 1D to 2D can be resolved by proper pairing of boundary filter types for decomposition. It is shown in section 5.4.6 how Type I, II and III boundary filters can be used to analyze signals of arbitrary lengths without redundancies. We further explain how the choice of these filter types can affect the balance of low and high-pass coefficients in a decomposition. It is demonstrated that by using multiscale edge locations to predetermine interval lengths at different scales, a non-expansive and balanced decomposition can be obtained. Additionally, different pairs of boundary filter types can help align the interior functions between consecutive rows of data in order to avoid reconstruction artifacts. Experiments with non-linear image approximation show that the proposed 2D interval wavelets decomposition method results in improved sharpness around image

edges as compared to standard wavelet techniques.

In this chapter, we see that interval wavelet transform outperforms standard wavelet transform for piecewise smooth signals. Essentially, the signals or images are analyzed in piecewise manner, which gives improved reconstruction in the vicinity of sharp edges. In the proposed embedded perceptual coding model (see 1.5 and 1.6), the images from the L2 layer are expected to be piecewise smooth. Therefore, the interval wavelet analysis is a suitable tool for the coding framework.

Algorithm 6.1: IntervalDWT

Input: Original interval $S = \{x_k\}_{e_0 \le k \le e_1}$, such that $S \subset X$.
Input: Low-pass interior filter H.
Input: High-pass interior filter G.
Output: Low-pass coeff. $C = \{c_k\}_{0 \le k < \lceil (N-1)/2 \rceil}$, where $N = e_1 - e_0 + 1$.
Output: High-pass coeff. $D = \{d_k\}_{0 \le k < \lfloor (N-1)/2 \rfloor}$, where $N = e_1 - e_0 + 1$.

Remark: Type-{A,B} denotes Type-A and Type-B filter for *left* and *right* boundary respectively.

begin
  $C \longleftarrow \emptyset$;
  $D \longleftarrow \emptyset$;
  if $S = \emptyset$ **then**
    /*interval length is 0 */
    return(C, D);
  end
  if $N = 1$ **then**
    /*Interval length is 1, sequence not filtered. */
    /*Copy sequence to next coarser scale. */
    if $\lfloor e_0/2 \rfloor = \lfloor e_1/2 \rfloor$ **then**
      $C \longleftarrow \{\sqrt{2} x_{e_0}\}$;
    else
      $D \longleftarrow \{\sqrt{2} x_{e_0}\}$;
    end
  else
    /*Determine F, the filter set to use on the left and right boundaries. */
    if *(e_0 even) and (e_1 odd)* **then**
      $\{H^{\text{left}}, H^{\text{right}}, G^{\text{left}}, G^{\text{right}}\} \longleftarrow$ Type-{I,I} boundary filter;
    else if *(e_0 odd) and (e_1 odd* **then**
      $\{H^{\text{left}}, H^{\text{right}}, G^{\text{left}}, G^{\text{right}}\} \longleftarrow$ Type-{II,I} boundary filter;
    else if *(e_0 even) and (e_1 even)* **then**
      $\{H^{\text{left}}, H^{\text{right}}, G^{\text{left}}, G^{\text{right}}\} \longleftarrow$ Type-{I,III} boundary filter;
    else
      $\{H^{\text{left}}, H^{\text{right}}\} \longleftarrow$ Type-{II,III} boundary filter;
    end
    $S_{pre} \longleftarrow$ `PreCondition` (S);
    $C \longleftarrow \left[H^{\text{left}}; H; H^{\text{right}}\right] S_{pre}^{T}$;
    $D \longleftarrow \left[G^{\text{left}}; G; G^{\text{right}}\right] S_{pre}^{T}$;
    $C \longleftarrow$ `PostCondition` (C);
  end
  return(C, D);
end

Algorithm 6.2: 2DBoundaryWT

Input: Original $M \times N$ image X.
Input: Number of decomposition levels, L.

```
1  begin
       /*Binary Edge detection on original image X                    */
2      {E_v, E_h} ⟵ BinaryEdgeDetect (X);
       /*Edge jitter correction                                       */
3      E_v ⟵ EdgeJitterCorrection(E_v);
4      E_h ⟵ EdgeJitterCorrection(E_h);
       /*Obtain multiscale vertical edge locations                    */
5      Ẽ_v ⟵MultiScaleEdgeLoc (E_v, L);
       /*Obtain multiscale horizontal edge locations                  */
6      Ẽ_h ⟵MultiScaleEdgeLoc (E_h, L);
       /*Perform L-level decomposition                                */
7      C ⟵ X;
8      for scale s ← L to 1 do
9          foreach row i ← 0 to M − 1 do
               /*Extract interval S using info. in Ẽ_v               */
10             foreach interval S in row i do
11                 IntervalDWT (S);
12             end
13         end
14         foreach column j ← 0 to N − 1 do
               /*Extract interval S using info. in Ẽ_h               */
15             foreach interval S in column j do
16                 IntervalDWT (S);
17             end
18         end
19     end
20 end
```

7 Perceptual Image Coding II

We don't see things as they are, we see things as we are. —Anaïs Nin (1903 - 1977)

Both standard and boundary wavelet techniques still treat images simply as a matrix of pixel values — a rather un-natural representation of images. Objects in images are not perceived as a collection of points of different colors and intensity, but at a macroscopic level as lines, shapes, contours etc. Due to their visual and perceptual importance, object boundaries are of particular interest to observer These types of boundaries tend to possess certain geometric properties such as direction, continuity and curvature order and standard wavelets are unable able to handle these macroscopic features efficiently. As discussed in previous chapters, standard wavelets representation are not efficient for signal representation across and along the boundaries. For example in 2D decomposition, many fine scale wavelets to represent a simple arbitrary-orientated line since the wavelets contains only horizontal, diagonal and vertical orientations. It would be more efficient if the geometrical information of the image boundaries can be extracted and coded. New representation methods must be sought. This lead to the proposal of boundary wavelets and their applications in chapters 4, 5 and 7, that seeks to yield sparser representation by not filtering across image boundaries. However, boundary

wavelet decomposition requires knowledge of boundary locations which have to be detected and entropy-coded as side information for proper reconstruction. Alternatively, this boundary information can be represented by objects known as *wedgelets*.

In this chapter, we review the basic concepts and application of wedgelets for compact image representations. Algorithms to reduce coding overheads for wedgelet parameters are also proposed. Additionally, the limitations and inefficiencies of the original wedgelets are demonstrated and a new multi-layered wedgelet is proposed in section 7.2 to overcome these problems. The techniques discussed in this chapter will be appropriate for encoding data in L1 layer in our proposed embedded perceptual coding framework discussed in chapter 1.

7.1 Wedgelet Analysis

The idea of analyzing images at different scales is both an intuitively natural and powerful approach. Examples of analysis based of multiscale framework are *gaussian pyramid*, *laplacian pyramid* and *dyadic wavelets*. Following the same principle, Donoho *et al.* [109][37] introduce 2D objects called *wedgelets* and *beamlets*, which are essentially localized 'functions' at different scales, locations and orientation that will give piecewise linear approximations of image contours and segments. Like wavelets, both wedgelets and beamlets are used to decompose an image at different dyadic scales. However, wedgelets and beamlets decomposition is fundamentally different from waveform-based transform coding. In the following, we briefly illustrate the main concept behind wedgelets and their associated approximations.

First, some terminology and notation. A dyadic partition or square S is a collection of points $\left\{(x_1, x_2) : [2^{-j}k_0, 2^{-j}(k_0+1)] \times [2^{-j}k_1, 2^{-j}(k_1+1)]\right\}$ where $0 \leq k_0, k_1 < 2^j$ for integer $j \geq 0$. Clearly, $S \subset [0,1]^2$. Denote $S_{j,\vec{k}}$ as dyadic squares at scale 2^j and position $\vec{k} = [k_0, k_1] \in \mathbb{Z}^2$. Thus, $S_{0,\mathbf{0}}$ is the unit square $[0,1]^2$. A formal description of wedgelet is given by:

Definition 7.1.1. *Let $\mathcal{L}$ be a fixed set of straight lines in $\mathbb{R}^2$. A wedgelet is obtained from a dyadic partition S of image I by dividing elements of S along lines belonging to $\mathcal{L}$.*

The set of line segments $\mathcal{L}$ are also called *edgel* in computer vision literature. Each

line segment $e \in \mathcal{L}$ can be denoted by $\overline{v_0 v_1}$, where v_0 and v_1 are vertices on the boundary of S. To find a wedgelet approximation of an image, it is necessary to compute v_0 and v_1 for each wedgelet partition. Consider the collection of all dyadic partitions at scales $0 \leq j \leq J$, let the smallest inter-vertex distance on the partition boundaries be $\delta = 2^{-J}$. The collection of all lines connecting vertices on the boundary of S_j is given by:

$$E_\delta(S_j) = \left\{ e = \overline{v_0 v_1} : 0 \leq v_0, v_1 < M_j \right\}, \tag{7.1}$$

where $M_j = 4(2^{-j}\delta)$ is the total number of vertices on the partition perimeter and there are a total of $\begin{pmatrix} M_j \\ 2 \end{pmatrix}$ possible line segments. Divided by a line e through S, a wedgelet consist of two *wedges*, or segments R_0 and R_1 (see figure 7.1(a)) with level constants (c_0, c_1) calculated by the averaging over appropriate region of image f:

$$c_0 = E(f(S_{j,\vec{k}})|R_0), \tag{7.2}$$

$$c_1 = E(f(S_{j,\vec{k}})|R_1), \tag{7.3}$$

Thus each wedgelet is parameterized by a quadruple (v_0, v_1, c_0, c_1). If the image I is M palette-indexed, the level constant c can be derived alternatively [110] as:

$$\min_{c \in \{0,\ldots,M-1\}} \left\{ \sum_{(x,y) \in \{S_{j,\vec{k}}|R_n\}} \delta(c - f_{x,y}) \right\}. \tag{7.4}$$

An overcomplete dictionary of wedgelets of various scales j and orientation $\overline{v_0 v_1}$, together with a *degenerated* function that is constant over S can synthesize any arbitrary image. The general idea of wedgelet decomposition is to partition a 2D image[1] f on the unit square $[0,1]^2$ into disjoint sets, followed by an wedgelet approximation of each partition. For each image partition $f(S_{j,k_0,k_1})$, we choose a wedgelet w that minimizes the

[1] In general, image f is a function of continuous variable on the unit square. For discrete $N \times N$ images, f is piecewise constant over squares of dimension $1/N$.

mean-square error by

$$W(f(S_{j,\vec{k}})) = \min_{v_0,v_1,c_0,c_1} \|f(S_{j,\vec{k}}) - w_{j,\vec{k}}(v_0, v_1, c_0, c_1)\|_{l_2}, \tag{7.5}$$

Then a multiscale wedgelet decomposition W of an image f is given by the collection $\mathcal{W}$ of partition wedgelet approximation of various scales,

$$\mathcal{W} = \left\{ W(f(S_{j,\vec{k}})) \quad : \quad j = 0, 1, \ldots, J-1 \quad k_0, k_1 = 0, 1, \ldots, 2^j \right\}. \tag{7.6}$$

Nevertheless, the collection is redundant. One need to select, for some set $\mathcal{J}$ and $\mathcal{K}$, an appropriate set of wedges $\{w_{j,\vec{k}}\}$ such that

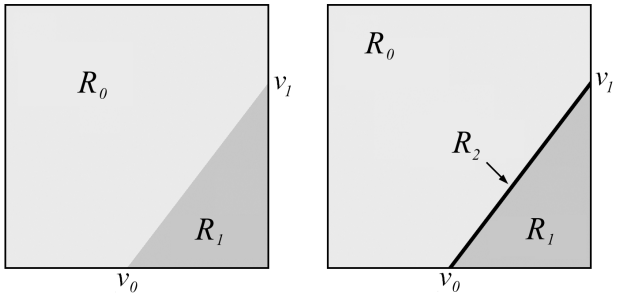
$$\bigcup_{j\in\mathcal{J},\vec{k}\in\mathcal{K}} S_{j,\vec{k}} = [0,1]^2 \quad \text{and} \quad \bigcap_{j\in\mathcal{J},\vec{k}\in\mathcal{K}} S_{j,\vec{k}} = \emptyset.$$

Each wedgelet can succinctly represent a straight contour. Smooth contours can be approximated by concatenating individual wedgelets. In places where the contour varies slowly, an accurate representation can be obtained using a few coarse scale wedgelets. Finer scale wedgelets are used where the contour varies more quickly. The major strength of wedgelet analysis is that it is able to capture the geometrical structures of an image at different scales. Gross geometric shapes can be inferred from coarse approximations. Thus, refining the analysis scale will result in both better geometrical representation and improved approximation accuracy.

Given a wedgelet decomposition $\mathcal{W}$ and its synthesized image g, the wedgelet approximation of an image f minimizes the functional

$$H_{\gamma,f}(g, \mathcal{W}) = \|f - g\|_2^2 + \lambda K(\mathcal{W}), \tag{7.7}$$

where K is some penalty function which could simply be the number of wedges, i.e., $K(\mathcal{W}) = |\mathcal{W}|$. λ is a regularization parameter. For $\lambda = 0$, the minimization algorithm will return $g = f$. As λ tends to $+\infty$, the minimizer is a constant image. The minimizer of eqn. (7.7) is denoted as $(\hat{g}_\lambda, \hat{W}_\lambda)$ where $\hat{W}_\lambda$ is an optimal partition, which gives the



(a) Wedgelet (b) Beamlet+Wedgelet

Figure 7.1: Examples of wedgelet [37] and beamlet+wedgelet [110] objects. The line $\overline{v_0 v_1}$ divides the partition into two wedges or segments R_0 and R_1, each parameterized by a level constant c_0 and c_1 respectively. The beamlet+wedgelet has a additional segment R_2, i.e., the beamlet.

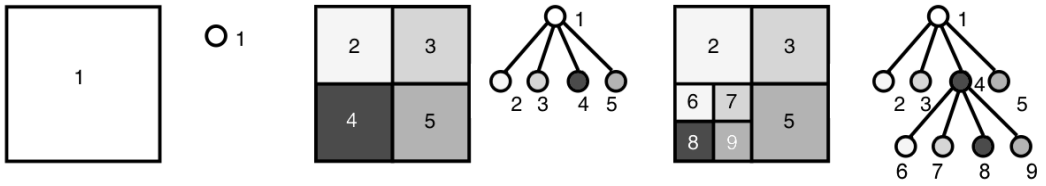


Figure 7.2: 2D examples of recursive dyadic partition tree. Example codewords for the trees are, from left to right, {END}, {SPLIT, END, END, END} and {SPLIT, END, END, SPLIT, END, END, END, END, END}.

corresponding image synthesis $\hat{g}_\lambda$ that minimizes the norm distance $\|f - g\|_2^2$ for a fixed λ. In the following, we see how a tree structure can be exploited to obtain $(\hat{g}_\lambda, \hat{W}_\lambda)$.

7.1.1 Tree Representation

To obtain an efficient wedgelet analysis, a *recursive dyadic partition* (RDP) tree structure is necessary to represent the decomposition in eqn. (7.6) for partitions at various scales and spatial locations. The RDP is well-known in time-frequency analysis literature. Consider the 1D case:

Definition 7.1.2. *The set of* recursive dyadic partitions *(RDP) on unit interval* $[0, 1]$ *are partitions* $\mathcal{P}$ *constructed with the following rules:*

- *The trivial partition* $\mathcal{P} = \{[0, 1]\}$ *is an RDP.*
- *If* $\mathcal{P} = \{S_1, \ldots, S_i, \ldots, S_m\}$ *is any existing RDP, then the partition obtained by half-splitting any* S_i *into adjacent dyadic intervals is also an RDP.*

For example, the partition $\mathcal{P} = \{[0, \frac{1}{2}), [\frac{1}{2}, \frac{3}{4}), [\frac{3}{4}, 1)\}$ is an RDP of unit square whereas the partition $\mathcal{P} = \{[0, \frac{1}{4}), [\frac{1}{4}, \frac{3}{4}), [\frac{3}{4}, 1)\}$ is not. Using the half-splitting rule (definition 7.1.2), we can denote the split partitions of S_k as $\{S_{k,0}, S_{k_1}\}$, where $S_k = S_{k,0} \cup S_{k,1}$. In this manner, $S_{k,0}$ and S_{k_1} can be considered as children of S_k in a binary tree relation. The set of RDPs related such that they form a RDP tree. For 2D partitions, the RDPs are defined over $[0, 1]^2$ and the quad-splitting rule applies. Thus, the quad-tree can be viewed as a roadmap that describes how to produce a given RDP tree by applying quad-splitting rules starting from the unit interval $[0, 1]^2$. Conversely, one can begin with a *complete* RDP tree, i.e., all terminal nodes are at the same depth, and apply quad-merging rule to the terminal nodes, in other words, *pruning* the tree. To code the entire RDP tree structure, it is sufficient to use only a set of two-symbol codewords, denoting whether to *split* ('SPLIT') or to *terminate* ('END') a node. Figure 7.2 shows some examples of RDP trees and their respective parsing codewords.

7.1.2 Wedgelet Approximation

With the defined RDP tree, we can have an efficient representation of wedgelet decomposition by assigning wedgelet parameters to each leaf node. A uniform scale wedgelet analysis will generate a complete RDP tree with all terminal nodes at the same depth. However, a uniformly partitioned representation is not efficient since the wedgelets are not exploited to adapt to slow varying contours using coarser scale representations. Therefore, a complete RDP tree has to be *pruned* in order to achieve a more compact and efficient tree representation. By pruning a RDP tree, we are trying to obtain a desired compression and quality trade-off — a severely pruned tree will lead to high compression rate but poor image reconstruction quality. Using a variation of eqn. (7.7), the objective is to find a pruned RDP tree $\mathcal{T}$ representation of the wedgelet decomposition such that function G is minimized:

$$G(\lambda) = \min_{\mathcal{T}} \Big(\|f - g(\mathcal{T})\|_{l_2} + \lambda C_1(\mathcal{T}) + \lambda C_2(\mathcal{T}) \Big), \quad \lambda \in \mathbb{R}^+, \tag{7.8}$$

where $g(\mathcal{T})$ is the reconstructed image from the pruned RDP tree $\mathcal{T}$. $C_1(\mathcal{T})$ denotes the bit cost of encoding the RDP tree $\mathcal{T}$. If the RDP tree is entropy-coded using the Huffman

technique, then, assuming a M-symbols RDP tree, C_1 can be estimated by

$$-\sum_{i=0}^{M-1} N_i \log_2 \frac{N_i}{|\mathcal{T}|}, \tag{7.9}$$

where N_i is the i^{th} symbol count and $|\mathcal{T}|$ is the total number of terminal and intermediate nodes on the RDP tree. $C_2(\mathcal{T})$ is the bit cost of coding the wedgelet parameters for all terminal nodes on $\mathcal{T}$. The bit cost for each $N \times N$ partition or terminal node is given by

$$N_v \log_2 4(N-1) + N_c \log_2(N_p) \tag{7.10}$$

where N_v and N_c is the number of orientation and level constant parameters for each partition, and N_p is the total number of color entries for palette images or total number of possible RGB colors representable in the decomposition.

To obtain a $\mathcal{T}$ that minimizes eqn. (7.8), consider a simple quad-tree, with a parent node with wedgelet w_{j-1} and four children wedgelets $\{w_{j,0}, w_{j,\vec{k}+(2^{-j},0)}, w_{j,\vec{k}+(0,2^{-j})}, w_{j,\vec{k}+(2^{-j},2^{-j})}\}$. Let MSE(.,.) denotes the mean square error measure between two functions. If the following inequality if true,

$$\begin{aligned}\frac{1}{4}\sum_{x=0}^{1}\sum_{y=0}^{1} \mathrm{MSE}\left(I(S_{j,\vec{k}+2^{-j}(x,y)}), w_{j,\vec{k}+2^{-j}(x,y)}\right) > \\ \mathrm{MSE}\left(I(S_{j-1,\vec{k}}), w_{j-1,\vec{k}}\right) - 4\lambda\end{aligned} \tag{7.11}$$

then the children are pruned and the parent node becomes the terminal node on the branch. A trivial case occurs when all the children nodes w_j are degenerated functions, and the inequality is automatically true. Algorithm 7.1 summarizes the steps for wedgelet approximation of an image.

7.1.3 Digital Wedgelets

When working with digital images, a discretized form of wedgelet analysis is required. There exist two major problems. First, a definition of digital lines is needed. The second problem, which follows from the first, is the discrete angular resolution of the digital lines. As in the continuous domain, digital wedge partitions are obtained

Algorithm 7.1: Wedgelet Approximation

Input: Original $N \times N$ image X.
Input: Number of decomposition levels, J.
Input: Regularization parameter. λ.

```
1  begin
       /*Computer wedgelet decomposition of image X for each scale                    */
2      for scale j ← J − 1 to 0 do
3          W_j = ∅;
4          for k_0 ← 0 to 2^j do
5              for k_1 ← 0 to 2^j do
6                  k⃗ ← (k_0, k_1);
7                  W_j ← {W_j, W(X(S_{j,k⃗}))}; /*see eqn. (7.5)                     */
8              end
9          end
10     end
       /*Create a complete RDP tree from J scales of wedgelet decomposition.        */
11     T ⟵ CreateRDPTree ({W_0, W_1, ..., W_{J−1}});
12     T̃ ⟵ PruneRDPTree (T, λ);
13     return(T̃);
14 end
```

by splitting dyadic partitions along straight lines. The study of *discrete radon transform* [111][112] has provided some good guidelines of discretizing of lines for the digital domain. Generally, the normal form of straight line equations $L_{d,\theta}$ are used, parameterized by the angle of the slope $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ and the perpendicular distance d from the origin. The pixels within a certain distance of the line are denoted as elements of the line. The discrete lines should be formulated such that each pixel p belongs to only one line for each θ, i.e.,

$$L_{d_1,\theta} \cap L_{d_2,\theta} = \emptyset, \quad d_1 \neq d_2.$$

This will avoid pixel-to-line assignment ambiguities. Using the same principal, Führ [113] defined a discretized line as follows:

Definition 7.1.3. *Let $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2})$ be given and define $v_\theta^\perp = (-\sin\theta, \cos\theta)$. Moreover, let*

$$\delta = \max\{|\sin\theta|/2, |\cos\theta|/2\}. \tag{7.12}$$

The digital line through the origin in direction v_θ is then defined as

$$L_{0,\theta} = \{p \in \mathbb{Z}^2 : -\delta < \langle p, v_\theta \rangle \leq \delta\}, \tag{7.13}$$

for some small δ. Moreover, define $L_{n,\theta}$ for $n \in \mathbb{Z}$ as

$$L_{n,\theta} = \begin{cases} \{p + (n, 0) : p \in L_{0,\theta}\}, & |\theta| > \frac{\pi}{4} \\ \{p + (0, n) : p \in L_{0,\theta}\}, & |\theta| \leq \frac{\pi}{4} \end{cases} \tag{7.14}$$

The set $\{L_{n,\theta}\}_{n\in\mathbb{Z}}$ partitions $\mathbb{Z}^2$ such that

$$\mathbb{Z}^2 = \bigcup_{n\in\mathbb{Z}} L_{n,\theta}, \tag{7.15}$$

and

$$L_{n,\theta} \cap L_{m,\theta} = \emptyset, \quad n \neq m. \tag{7.16}$$

Hence, for a given orientation θ, we can have a set of wedge splits (i.e., R_0 and R_1 as in figure 7.1(a)) by linear translation n. We have not mentioned how to discretize the angular resolution for θ. It is inefficient to use a fixed set Θ of angles to partition dyadic squares of various dimensions or scales. For small dyadic squares, a small angular step Δ can lead to $L_{n,\theta} = L_{n,\theta+\Delta}$, thus yielding identical wedgelets. Generally, a dyadic square of size 2^j can resolve $O(2^j)$ angles. One can predetermine a finite set $\Theta \subset [-\frac{\pi}{2}, \frac{\pi}{2})$ of admissible angles for each partition partition dimension $N = 2^j$, thus resulting in a scale dependent angular resolution which can, additionally, reduce complexity of wedgelet splitting computation. Finally, the discrete wedgelet on a square partition can be defined as follows:

Definition 7.1.4 (Führ, Demaret, Friedrich)**.** *Let $q \subset \mathbb{Z}^2$ be a square and for $(n, \theta) \in \mathbb{Z}^2 \times [-\frac{\pi}{2}, \frac{\pi}{2})$, let $L(q, \theta)$ denote the set of discrete lines $L_{n,\theta}$ such that $L_{n,\theta} \cap q \neq \emptyset$ and $L_{n+1,\theta} \cap q \neq \emptyset$. The discrete wedge splitting is the partition of q in two wedges $\{w^1_{n,\theta}(q), w^2_{n,\theta}(q)\}$ defined by*

$$w^1_{n,\theta}(q) = \bigcup_{k\leq n} L_{k,\theta} \cap q$$

and

$$w^2_{n,\theta}(q) = \bigcup_{k>n} L_{k,\theta} \cap q.$$

The set of two wedges $\{w^1, w^2\}$ forms a wedgelet.

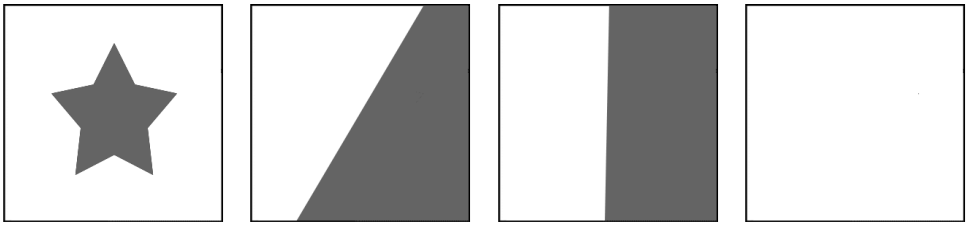
7.1.4 Fast Wedgelet Decomposition

To estimate the orientation parameters v_1 and v_2, each $N \times N$ partition requires computing $\frac{1}{2}(N-1)(N-2)$ possible projections – an order of $O(N^2)$. Moreover, each projection require N^2 operations thus leading to $O(N^4)$ computations for each partition approximation. In applications where N is large, the order of complexity becomes impractical for wedgelet decomposition.

Nevertheless, there exist several techniques to reduce the computational complexity of wedgelet decomposition. Observe that the ratio of operations required for a partition at scale $j-1$ to a quad partition set at scale j is given by $4:1$. As a consequence, it is possible to reduce the computation cost by analyzing at fine scales and then obtain an estimation for the coarser scales. An example of such exploitation is found in [114] whereby inter-scale relationships between wedgelets profiles are utilized to check for admissible wedgelet orientations – thus, effectively narrowing the search space for optimal wedgelet approximation. Alternatively, it is hinted in [37] that it is possible to approximate a wedgelet analysis in no more than $O(N^2 \log N)$ flops. For example in [115], the integral over the image partition are precalculated such that no more than six operations are needed for the sum $\sum_{(x,y)\in w} f(x,y)$ for every wedge w on the partition.

7.1.5 Partition Bounded Segments

A wedgelet consists an edge contour that intersects the partition boundary. This becomes an issue when we attempts to approximate a partition that contains a bounded segment (see figure 7.3(a)). Both figure 7.3(b) and 7.3(c) are admissible solutions that minimize eqn. (7.5) but they give visually unpleasant results. A better solution is to represent the partition with a degenerated function, thus eliminating the bounded segment within (see figure 7.3(d)).



(a) (b) (c) (d)

Figure 7.3: (a) Original image with a partition bounded segment; (b)-(c) Possible wedgelet representations that minimize the mean-square error; (d) A degenerated type representation will eliminate the presence of the bounded segment, which is a more visually 'pleasing' and sensible solution.

The following outlines the procedures for eliminating the partition bounded segments using finer scale information:

1. Let w_{j-1} be a wedgelet function on partition S_{j-1}.
2. Let the quad-split partitions of S_{j-1} be $S_{j,0}$, $S_{j,1}$, $S_{j,2}$ and $S_{j,3}$, arranged in clockwise manner from top-left.
3. Let $\{w_{j,k}\}_{0\leq k<4}$ be a quad set of wedgelet functions on $\{S_{j,k}\}_{0\leq k<4}$.
4. Denote $L_{n,\theta}(w_j)$ as the wedge-split line for wedgelet w_j.
5. Let $B_t(S_j)$, $B_r(S_j)$, $B_b(S_j)$ and $B_l(S_j)$ denote, respectively, the top, right, bottom and left boundary line segments of partition S_j.
6. if ($L_{n_\theta}(w_{j,0})$ intersects $B_r(S_{j,0})$ and $B_b(S_{j,0})$) & ($L_{n_\theta}(w_{j,1})$ intersects $B_b(S_{j,1})$ and $B_l(S_{j,1})$) & ($L_{n_\theta}(w_{j,2})$ intersects $B_l(S_{j,2})$ and $B_t(S_{j,2})$) & ($L_{n_\theta}(w_{j,3})$ intersects $B_t(S_{j,3})$ and $B_r(S_{j,3})$), then let w_{j-1} be a degenerated function.

Figure 7.4 illustrates an example of segment elimination. Essentially, we check the orientation parameters $\vec{v}$ of the finer scale wedgelets. If none of them fall on the edge of the partition S_{j-1}, then a partition bounded segment exists and is eliminated by declaring a degenerated function on S_{j-1}.

7.1.6 Excessive Fine Partitions

As mentioned in section 7.1, fine scale wedgelets are intended to represent contours of high curvatures. However, there are exceptions whereby simple linear contours can

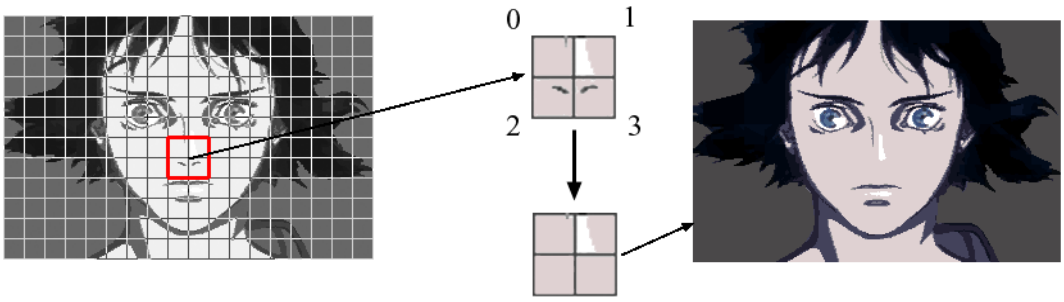


Figure 7.4: Segment elimination example.

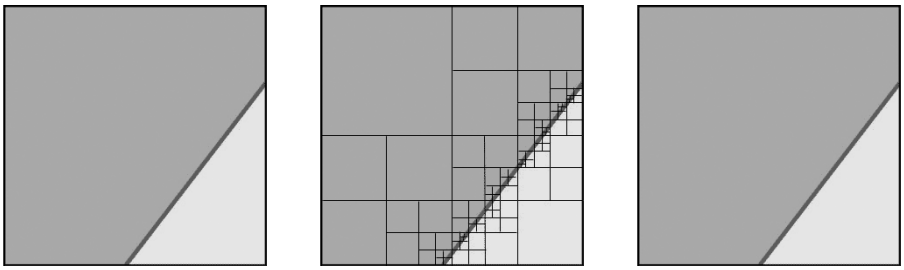


Figure 7.5: (a) The original image partition with a colored line and two piecewise constant segments. (b) Decomposition with only wedgelet result in very fine scale partitions in the vicinity of the line segment. (c) Decomposition with beamlet+wedglet type requires only one scale representation.

result in excessive fine partitioning. This leads to inefficient RDP tree representations. For example in figure 7.5(b), in order to obtain a visually good wedgelet approximation of narrow features, such as lines, many fine scale partitions are required for a simple linear structure. Therefore, in [110], we propose using beamlet+wedgelet combination (see figure 7.5) analysis to improve the coding performance. The main disadvantage of this technique is that the beamlet for a particular partition must have constant thickness. Similar excessive fine partitioning can also occur in regions that contain image *corners* or *junctions*. The inefficient wedgelet representation due to such features can be observed in examples given in figure 7.6(b) and 7.7(b). The following section proposes a solution to the fine partitioning problem using *multi-layered wedgelets*.

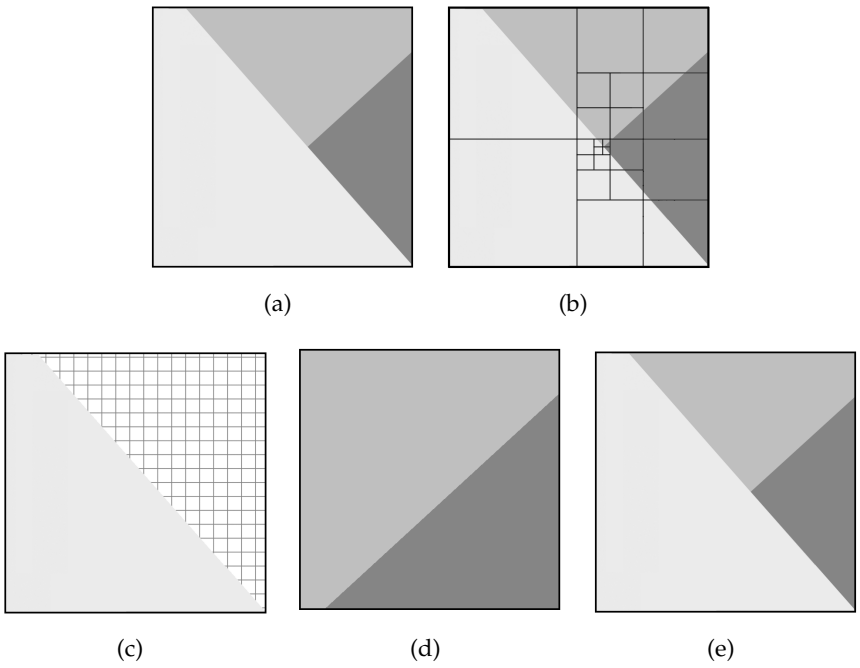


Figure 7.6: T-junction example. (a) Original image. (b) Partitions due to wedgelet analysis. For multi-layered wedgelet $\mathcal{W} = \{w_0, w_1, w_2\}$, wedges (c) w_0 and (d) $\{w_1, w_2\}$ are superimposed to obtain the final image partition in (e).

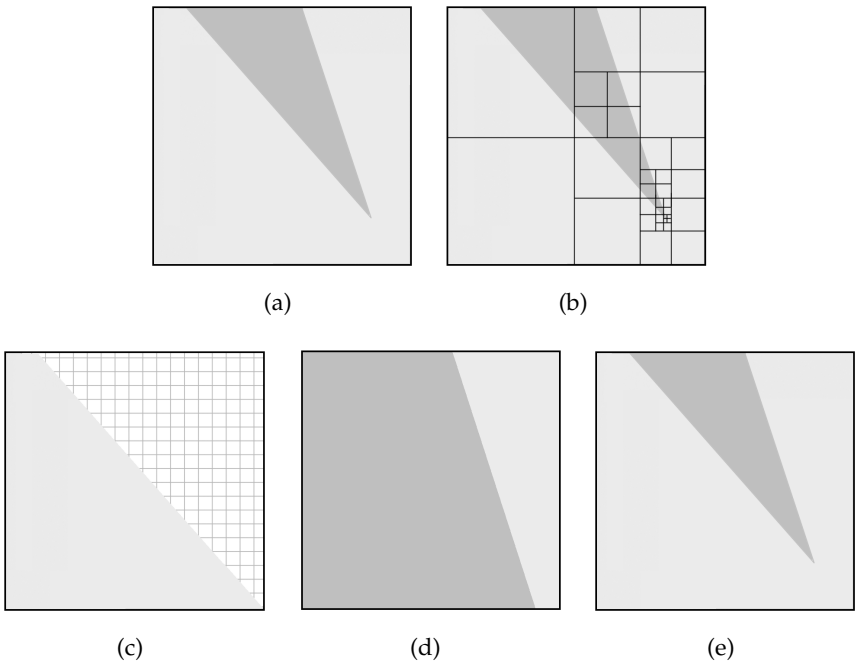
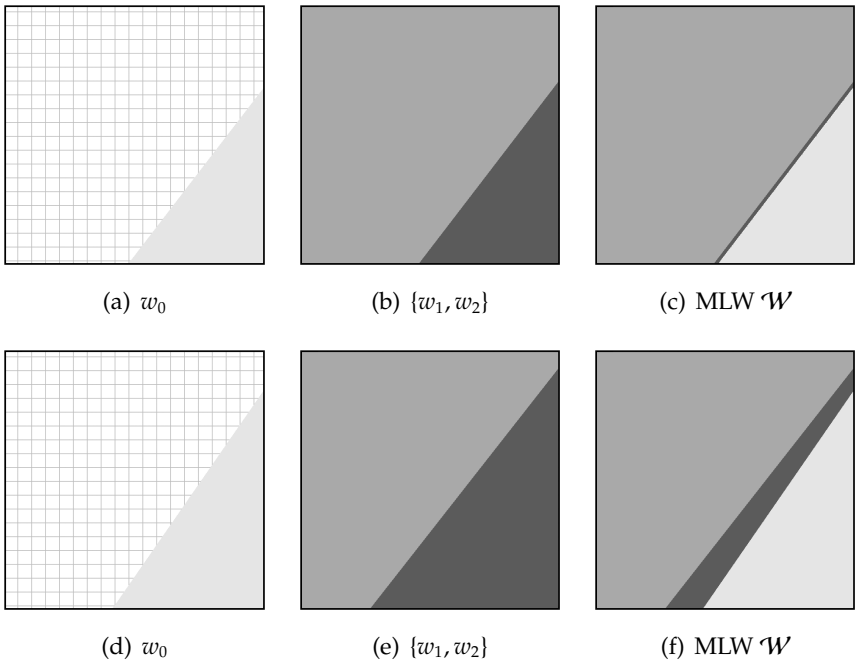


Figure 7.7: Corner example. (a) Original image. (b) Partitions due to wedgelet analysis. For multi-layered wedgelet $\mathcal{W} = \{w_0, w_1, w_2\}$, wedges (c) w_0 and (d) $\{w_1, w_2\}$ are superimposed to obtain the final image partition in (e).



(a) w_0 (b) $\{w_1, w_2\}$ (c) MLW $\mathcal{W}$

(d) w_0 (e) $\{w_1, w_2\}$ (f) MLW $\mathcal{W}$

Figure 7.8: Narrow feature examples. Multi-layered wedgelet $\mathcal{W} = \{w_0, w_1, w_2\}$ with wedges (a) w_0 and (b) $\{w_1, w_2\}$ superimposed to produce final image partition in (c). Alternate wedges (d) w_0 and (e) $\{w_1, w_2\}$ superimposed to produce a width-varying feature in (f).

7.2 Multi-Layered Wedgelet Analysis

The original wedgelet definition only allows one linear contour and two segments, or wedges, for each partition. In section 7.1.6, it is demonstrated that narrow features and segment junctions can result in excessive fine partitioning in wedgelet approximation. Consequently, this leads to inefficient representation due to a deep branching RDP tree with large number of parameters to be coded. To overcome this limitation, we propose superimposing several wedges on the same partition. This leads to new objects called *multi-layered wedgelets* or MLWs:

Definition 7.2.1. *For $L \geq 2$ and partition Q, a* multi-layered wedgelet *(MLW) $\mathcal{W}(L, Q) = \{w_1(Q, \bar{d}_1), w_2(Q, \bar{d}_2), \ldots, w_{L-1}(Q, \bar{d}_{L-1}), w_L(Q, \bar{d}_L)\}$ is an ordered superimposition of L wedges with quadruple $\bar{d} = (s, n, \theta, c)$, such that w_{l-1} overlays w_l. Moreover, $s_{L-1} = \bar{s}_L$, $n_{L-1} = n_L$ and*

$\theta_{L-1} = \theta_L$. *The wedge split for a partition of* Q *is given by*

$$w_{s,n,\theta}(Q) = \begin{cases} \bigcup_{k \leq n} L_{k,\theta} \cap Q, & s = 0. \\ \bigcup_{k > n} L_{k,\theta} \cap Q, & s = 1. \end{cases}$$

where $L_{k,\theta}$ *are discrete lines. The constant level parameter* c *of each wedge is obtained as*

$$c = E(Q | w_{s,n,\theta}(Q)).$$

A L-layered MLW will have up to $L-1$ contours and L segments on a partition. The original wedgelet is a special case of MLW for $L = 2$ where the union of the two wedges spans the partition. To illustrate the concept of MLW, we assume the case with $L = 3$ and thus $\mathcal{W}(3) = \{w_0, w_1, w_2\}$. In order for superimposition to work, we let $w_1 \cup w_2$ be a wedgelet and w_0 be a wedge with a different contour profile from w_1 and w_2. Furthermore, $\{w_k\}_{0 \leq k < 2}$ are ordered such that w_{k-1} overlays w_k. Using set operations, we can express the MLW for $L = 3$ as

$$w_0 \cup \{w_0^c \cap (w_1 \cup w_2)\}, \tag{7.17}$$

which allows a partition to be described by 2 piecewise linear contours and up to 3 segments. For the general case of L-layer MLW representation, each partition can have $L-1$ linear contours and up to L segments. The decomposition for a partition S is given by:

$$\mathcal{W}(L, I(S)) = \{w_0, w_1, \ldots, w_{L-1}\}, \tag{7.18}$$

such that the error-norm is minimized

$$\min_{\{w_0, \ldots, w_{L-1}\}} \| I(S) - w_0 - \sum_{p=0}^{L-2} w'_p \|_{l_2} \tag{7.19}$$

where

$$w'_p = \left(\bigcup_{q=0}^{p} w_q \right)^c \cap w_{p+1}. \tag{7.20}$$

Eqn. (7.20) implicitly defines the superimposition order of the wedges $\{w_k\}_{0 \le k < L}$. In figure 7.6 and 7.7, representations of image junctions and corners using MLW only require a single scale partition, whereas the standard wedgelet representations require deep-branching RDP trees into fine scales. MLW is also able to give an efficient single scale representation for narrow features as shown in figure 7.8. Unlike the beamlet+wedgelet combination method in [110], MLW provide a more robust representation of narrow features of arbitrary widths.

Theoretically, the upper limit of L is only restricted to the dimension of the partitions — more layers of wedgelets can be used for larger partitions. In practice, we let $L \le 4$ for practical computational reasons. Since the advantage of MLW analysis is to prevent excessive fine partitions or deep-branching RDP trees, it is not suitable for fine scale analysis, e.g. 2×2 pixels partition. For a complete RDP tree $\mathcal{T}$ with representation of MLWs, the tree pruning in section 7.1.1 applies where the penalty function K can be modelled simply as tree nodes count $|\mathcal{T}|$ or bit cost. The bit cost for each MLW on a $N \times N$ partition is given by

$$\log_2 L + 2L \log_2 3(N-1) + (L+1) \log_2 C. \tag{7.21}$$

where L is the number of layers or wedges, and C is the number of levels representable for each wedge w_k. The second term in eqn. (7.21) gives the total number of bits required to code the vertices of each wedge in the MLW. The last term is the number of bits for coding all the level constant information.

7.2.1 Erasing Wedge

The benefits of MLW is the ability to represent image junctions such as L-junction, T-junction and arrow-junction (see figure 7.9). However, it cannot give a single scale representation for Y-junctions, and X-junctions. The following proposition describes the limitation of the MLW representation:

Proposition 7.2.2. *A partition with L-layered MLW description, as defined in defn. 7.2.1, necessarily contains at least a segment without a corner or having a corner with interior angle greater than π radian.*

The limitation is demonstrated in figure 7.10(a) to 7.10(d) for the approximation of a X-junction where all the segments have interior angles less than π, and thus, it cannot be completely approximated by the MLW. Nevertheless, this limitation can be removed by introducing a new object that is called *erasing wedge*.

Definition 7.2.3. *An erasing wedge w_e subtracts elements p from another wedge w where $p \in w_e \cap w$.*

In the following, we demonstrate how the erasing wedge can be incorporated into the existing MLW framework. Consider a MLW, $\{w_0, w_1, \ldots, w_{L-1}\}$. If, for some $0 \leq k < L-1$, w_k is an erasing wedge the following applies:

- if w_k is an erasing wedge, then w_k removes elements from w_{k-1}.
- w_{L-1} cannot be an erasing wedge.

To obtain a final MLW representation whose elements consist of erasing wedges, let $\mathcal{U} = \{u_0, u_1, \ldots, u_{L-1}\}$ where for $0 \leq k < L$,

$$u_k = \begin{cases} w_{k+1} - w_k & \text{if } w_k \text{ is erasing wedge.} \\ 0 & \text{if } w_{k+1} \text{ is erasing wedge.} \\ w_k & \text{otherwise.} \end{cases} \tag{7.22}$$

The superimposition rule in eqn. (7.20) can be applied to $\mathcal{U}$ to obtain the final MLW. Figure 7.10(e) to 7.10(h) demonstrate a successful approximation of X-junction using MLW with an erasing wedge.

7.2.2 Fast MLW Decomposition

In section 7.1.4, it is shown that each wedgelet approximation has a computation complexity of $O(N^4)$. For a L-layer MLW, the computation may seem out of reach for exhaustive search of the L best matching wedges for each image partition. Since we

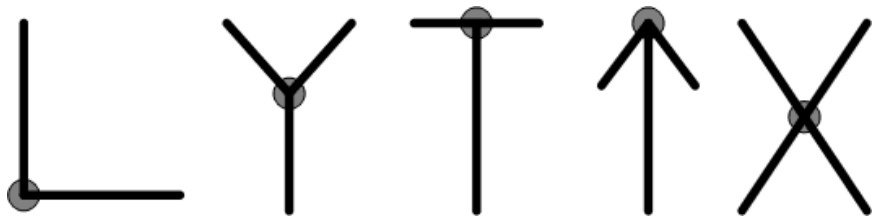


Figure 7.9: Junction and corner types. (*Left to right*): L-junction, Y-junction, T-junction, arrow-junction and X-junction.

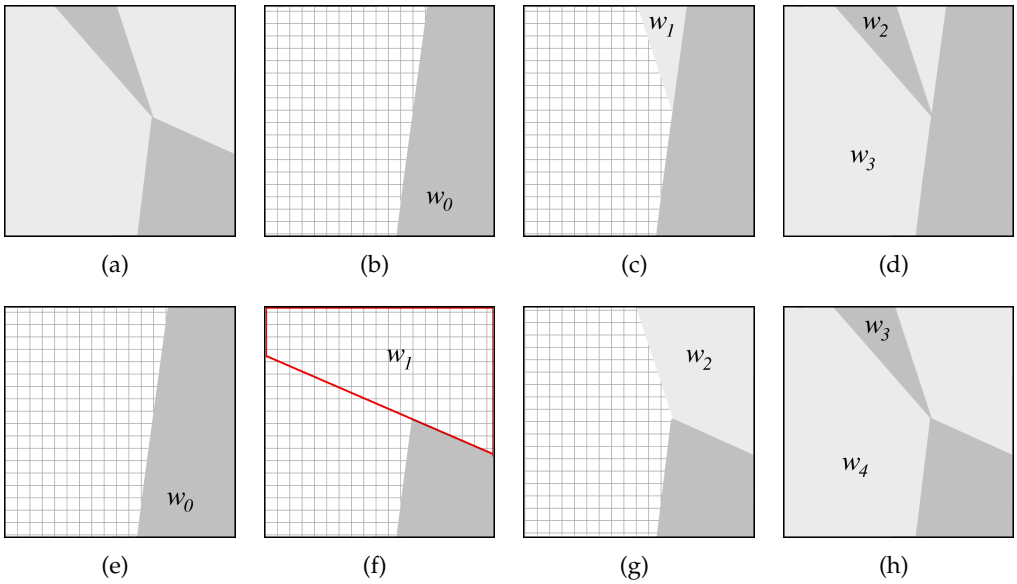


Figure 7.10: Approximation of a (a) X-junction using MLW (b)-(d) without erasing wedge and (e)-(h) with an erasing wedge w_1 with its boundary outlined in red. Note that the wedges $\{w_0, w_1, w_2, w_3\}$ in (b)-(d) are equivalent to the wedges $\{w_0, w_2, w_3, w_4\}$ in (e)-(h).

already have fast wedgelet decomposition solutions [114][115], we can prune an existing RDP tree of wedgelet decomposition by replacing branches by single leaf nodes of MLW representations. Thus the MLW partitions are created while pruning the RDP tree. First, we have the following proposition in which a coarser scale partition can also be inferred to have MLW from its children partitions.

Proposition 7.2.4. *Given a partition $S_{j,\vec{k}}$ and its finer scale dyadic quad-set $\{S_{j-1,\vec{k_0}}, S_{j-1,\vec{k_1}}, S_{j-1,\vec{k_2}}, S_{j-1,\vec{k_3}}\}$ such that $S_{j,\vec{k}} = \cup_{n=0}^{3} S_{j-1,\vec{k_n}}$. If at least one member in $\{S_{j-1,\vec{k_n}}\}_{0\leq n<4}$ is approximated by a MLW(L), then $S_{j,\vec{k}}$ is also approximated by a MLW with $L' \geq L$.*

If the children partitions contains only standard wedgelets, we consider whether a feasible representation can be obtained by merging them into a single partition of MLW at coarser scales through the following procedures:

1. Let $\mathcal{L} = \{L_{n_0,\theta_0}, L_{n_2,\theta_2}, \ldots, L_{n_3,\theta_3}\}$ be a set of line segments obtained from a quad set of wedgelet approximated partitions[2] at scale $j-1$.

2. Translate the normal equations of straight line in $\mathcal{L}$ to a partition at scale j, thus obtaining a new set $\mathcal{L} \rightarrow \tilde{\mathcal{L}}$.

3. Let $\mathcal{M}$ be empty set.

4. For each pair $\tilde{L}_{n_i,\theta_i}, \tilde{L}_{n_j,\theta_j} \in \tilde{\mathcal{L}}$ such that $i \neq j$,
   if $|n_i - n_j| < T_n$ and $|\theta_i - \theta_j| < T_\theta$ for some threshold $T_n, T_\theta > 0$,
   then let $\mathcal{M} \leftarrow L'_{a,b}$ where $a = (n_i + n_j)/2$ and $b = (\theta_i + \theta_j)/2$.

5. If $|\mathcal{M}| > 1$ then let partition at scale j be approximated by MLW using line segments from $\mathcal{M}$.

The resulting MLW at scale j will have $|\mathcal{M}|$ layers.

7.3 Tree Prediction

Given a wedgelet decomposition of an image, some partitions will be represented by degenerated wedgelets, denoted by only a level constant parameter. By exploiting

[2] assume these partitions have passed the *bounded segment elimination* test (see section 7.1.5).

the observation that the degenerated partitions are bounded by partitions containing wedges, it is possible to predict the level constant parameters of these degenerated functions. The following procedures outline the steps to predict the level parameters of partitions containing degenerated wedgelets.

1. Let I_1 be an image constructed from wedgelets.

2. Let I_2, be an image constructed from degenerated wedgelets only.

3. Define $\mathcal{R} = \{R_0, R_1, \ldots, R_{N-1}\}$, the N disjoint regions of I_2 such that

$$\bigcap_{i=0}^{N-1} R_i = \emptyset,$$

and

$$\bigcup_{i=0}^{N-1} R_i = I_2.$$

4. Let $B(R) = \{(x_p, y_p) : n = 0, \ldots, P-1\}$ be the set of coordinates on the outer perimeter of region R.

5. The predicted color $\hat{c}$ of R is given by,

$$\hat{c} = \begin{cases} \max_{c \in \{0,\ldots,M-1\}} \left(\sum_{(x,y) \in B(R)} \delta(c - I_1(x,y)) \right), & \text{if } I \text{ is } M\text{-palette image,} \\ \frac{1}{|B(R)|} \sum_{(x,y) \in B(R)} I_1(x,y), & \text{otherwise,} \end{cases} \tag{7.23}$$

where

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0, \\ 0, & \text{otherwise.} \end{cases}$$

6. If I is a M-palette image, goto step 8.

7. For each degenerated wedgelet w with color c, if $|c - \hat{c}| < \delta$ for some small δ, then label w as a predictable.

8. For each degenerated wedgelet w with color c, if $c = \hat{c}$, then label w as a predictable.

For degenerated wedgelets that are predictable, their level constant parameters are not coded.

7.4 Application: Cel Image Coding

Color distortion is a problem that is encountered when compressing cartoon images, which comprise of a foreground image called *cels* and a background image. Cel images generally have limited key colors, i.e., a fixed palette. When lossy wavelet or cosine transform coding techniques are applied to cel images, the reconstructed images will have colors outside the range of the original palette, which leads to observable color distortions. There are several methods to minimize this distortion, such as by re-quantizing/ re-mapping the decoded color image or by reordering color symbols [116][117] [118][119] prior to coding in order to minimize visual distance between neighboring color symbols. In contrast, color distortion problem can be reduced or avoided by using wedgelet representation since each wedge represents the mean color of the corresponding image region. Figures 7.11(a) and 7.11(b) give an illustrated overview of the wedgelet image coding and decoding system respectively.

7.4.1 Color reduction

Due to source noise, the wedgelet level parameters tends to differ slightly for regions that are originally of the same color. To increase parameter coding efficiency, we can merge colors that are spectrally close in the *hue-saturation-intensity* (HSI) space. It is noted that if two colors have a hue difference less than $\frac{\pi}{18}$, the human visual system cannot differentiate between them [118]. This color reduction process results in a sparse color palette to be coded for the image.

7.4.2 Parameter Coding

The structure of the RDP tree can be described by symbols {WEDGE, MLW, FLAT, PFLAT} for the terminal nodes denoting respectively,

- *wedgelet*, with parameters $\overline{v_0 v_1}$ and $\{c_0, c_1\}$.
- *MLW-wedgelet* (n-layers), with parameters n, $\{\overline{v_{0,0} v_{0,1}}, \overline{v_{1,0} v_{1,1}}, \ldots, \overline{v_{n-1,0} v_{n-1,1}}\}$ and $\{c_0, c_1, \ldots, c_{n-1}\}$.

- *degenerated wedgelet*, with parameter c_0,
- *predictable degenerated wedgelet*, no parameters.

The intermediate tree nodes are represented by the 'SPLIT' symbol. For palette images, each level constant parameters of the wedgelets are arithmetic coded and the color palette entries are binary uncoded[3]. The parameters $\overline{v_0 v_1}$ of the wedgelets are also binary uncoded; except for the 2×2 and 4×4 wedgelets, which will be arithmetic-coded.

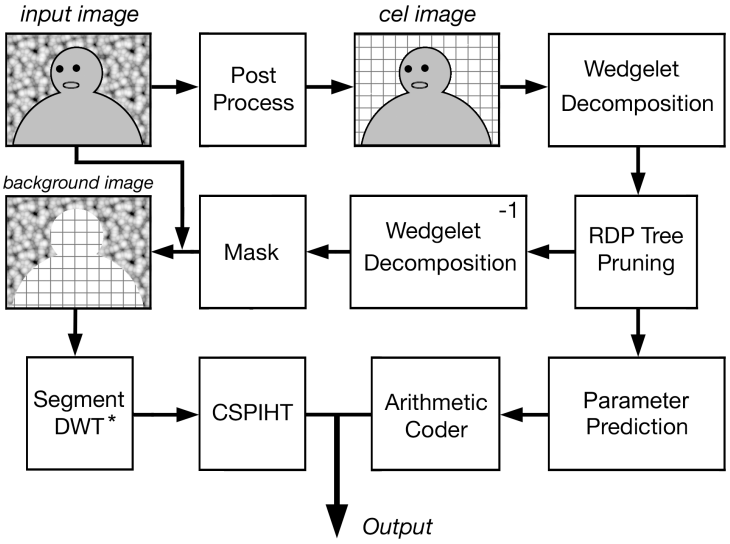
7.4.3 Background Image Coding

The background of cartoon image can be compressed using segment-based transform coding methods. In [110], the DWT coefficients of the segmented background image are modified using the *Projection Onto Convex Sets* (POCS) related techniques [120][121] such that the significant coefficients can be compacted within a certain segment mask and only coefficients within the mask are coded. Since we already have the segment boundary information coded in the wedgelet representation, we can alternatively interval wavelets transform the background image to yield a non-redundant expansion, i.e., the number of coefficients equals the number of elements in the mask. The resulting DWT coefficients are coded using CSPIHT (see chapter 3 and [122]). Unlike the original spatial orientation trees specified in SPIHT and CSPIHT, an equivalent subband masks based on cel/background masking information is used to obtain a new SOT configuration in which a parent node can have 1 to 4 possible children nodes.

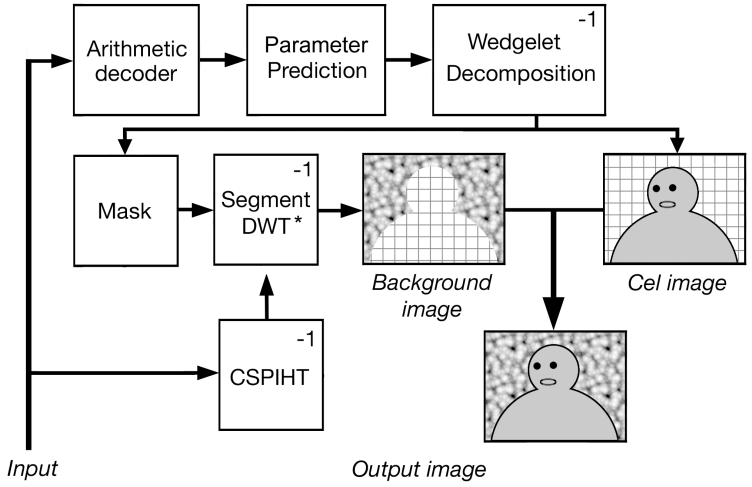
7.5 Numerical Examples

In figure 7.12, we demonstrate the advantage of using MLW in wedgelet analysis, which can leads to more efficient and visually improved representation. Figures 7.13 and 7.14 show the results of coding a cartoon image using JPEG2000 [21][22] and our proposed wedgelet technique. While the decoded JPEG2000 image in figure 7.13(b) and 7.14(b) contains the usual ringing artifacts along edge boundaries, compression artifacts are virtually absent in the wedgelet coded image in figure 7.13(c) and 7.14(c). However, detailed comparison with the original image will reveal that there are missing features in the wedgelet coded image.

[3] i.e., without any entropy coding



(a) Encoding



(b) Decoding

Figure 7.11: Overview of the cartoon image coding and decoding system.

Figure 7.17(b) and 7.17(c) shows an example of coding an image at an extremely low bitrate of 0.024bpp. Expectedly, the image produced by JPEG2000 is severely distorted with artifacts, making the image interpretation extremely difficult. When the same image is coded with wedgelets, all fine scale details and features are lost, while long-ranging contours representing marco structures are still visible and reasonably well defined. Since a color palette is used, there is no color distortion due to quantization errors even at high compression rates.

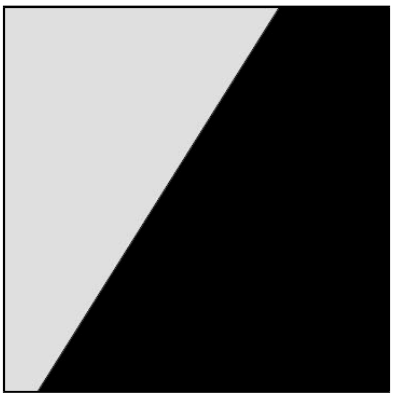
Our proposed coder/ decoder, which is designed to code animation cel type images, has also been tested with real images. In figure 7.19, even though the JPEG2000-coded image appears to have more details but it is highly distorted. In contrast, the wedgelet approximation in figure 7.19(b) provides a relatively more accurate and visually pleasant reconstruction of large structures and object boundaries even at very low bit rates. For very high compression, a good representation like cartoon sketches of the original image may be more preferable to visually-annoying artifacts produced by wavelet and cosine transform coding methods. Additionally, since the wedgelets contain geometric rendering information about the image explicitly, it is possible to improve the visual quality of the reconstructed image by post-processing methods such as anti-aliasing drawing, smooth connection of the contours and color interpolation or shading.

7.5.1 Summary and Remarks

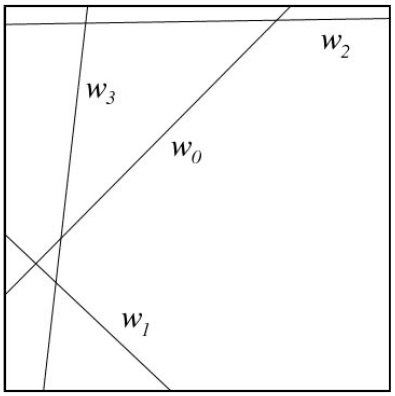
In this chapter, we have reviewed the wedgelet analysis technique that is suitable for detecting and representing geometrical structures in images, which is important for coding images in the L1 data layer of our proposed coding framework (see chapter 1). Several limitations of the original proposed wedgelet analysis introduced by Donoho *et al.* were observed. We proposed a new multi-layered wedgelet technique to improve the image approximation using wedgelet analysis. Regarding coding efficiency of wedgelet decomposition, we have also proposed a parameter prediction scheme for degenerated functions. In the numerical experiments with cel-based cartoon images, the proposed hybrid multiscale wavelet-wedgelet image coding scheme outperforms JPEG2000 in terms of visual quality. For very low bitrate simulations, wedgelet representations have

shown to be able to preserve macro features in the coded images that could still facilitate visual interpretation.

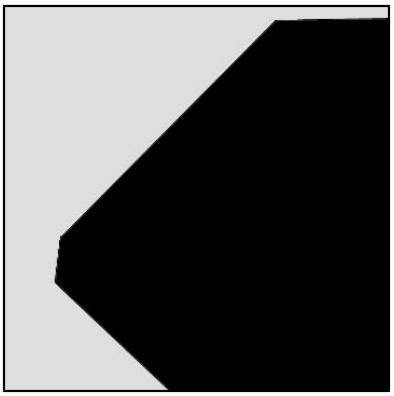
(a) Original



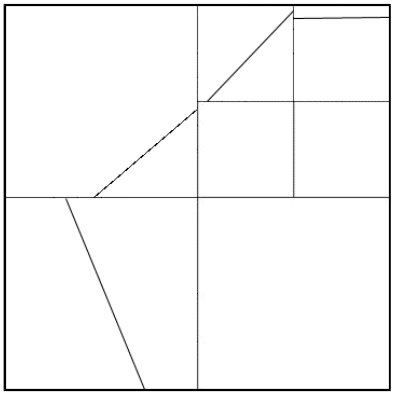
(b) Wedgelet approximation at single scale.



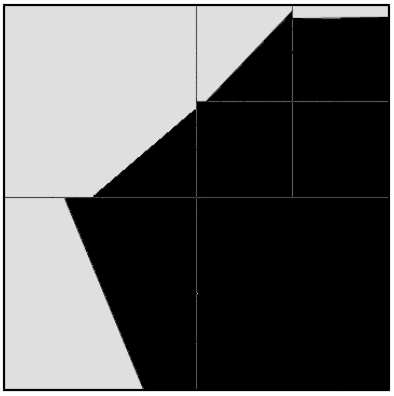
(c) MLWs



(d) MLW approximation



(e) Wedgelets



(f) Wedgelet approximation at two scales.

Figure 7.12: Using the *cameraman* image (zoomed and cropped), (a)-(d) compares wedgelet and multi-layered wedgelet ($L = 4$) using single scale approximation. (e)-(f) Wedgelet approximation with 4 wedges and 3 degenerated partitions at two scales.

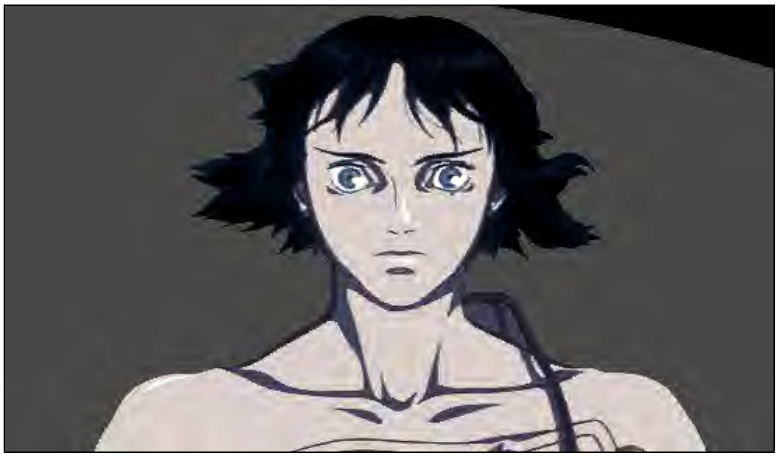
(a) Original.

(b) JPEG2000 at 0.29bpp.

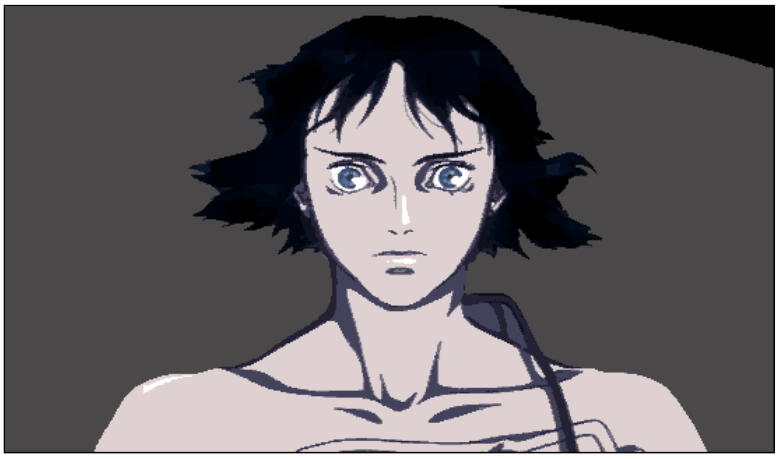
(c) Proposed Method at 0.29bpp.

Figure 7.13: A comparison between reconstructing a cartoon image.

(a) Original.



(b) JPEG2000 at 0.18bpp.



(c) Proposed Method at 0.18bpp.

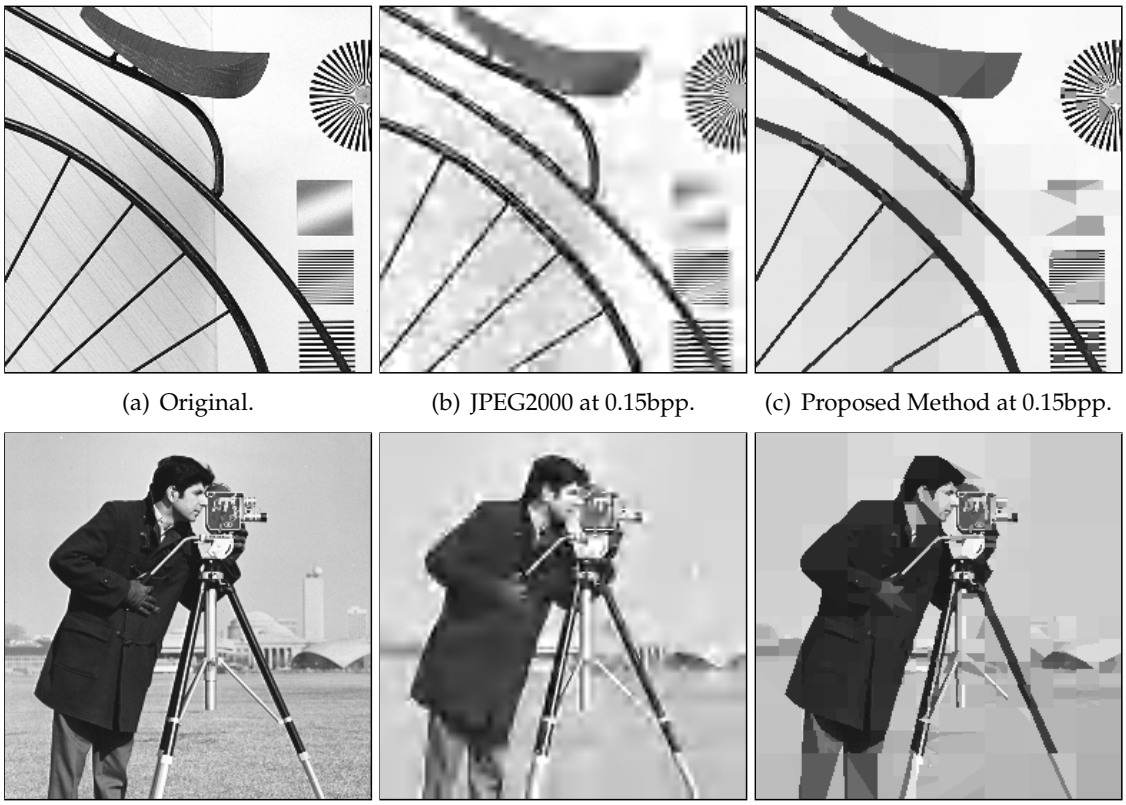
Figure 7.14: A comparison between reconstructing a cartoon image.

(a) Original.

(b) JPEG2000 at 0.22bpp.

(c) Proposed Method at 0.22bpp.

Figure 7.15: A comparison between reconstructing a cartoon image.



(a) Original. (b) JPEG2000 at 0.15bpp. (c) Proposed Method at 0.15bpp.

(d) Original. (e) JPEG2000 at 0.15bpp. (f) Proposed Method at 0.15bpp.

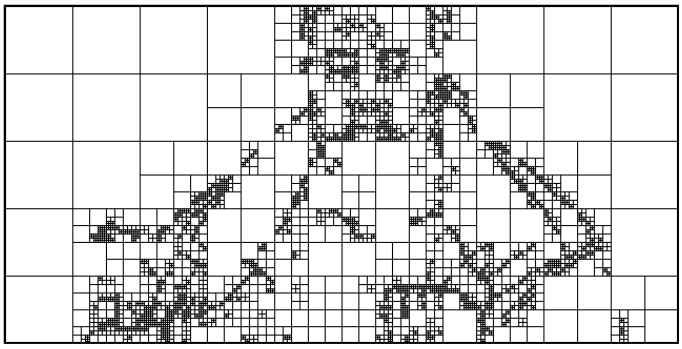
Figure 7.16: A comparison between reconstructing photographic images.

(a) Original

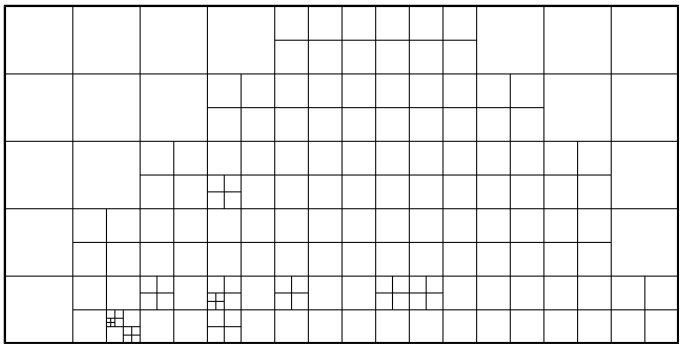
(b) JPEG2000 at 0.024bpp

(c) Proposed Method at 0.024bpp

Figure 7.17: A comparison between reconstructing an image at very low bit rate.



(a) Original



(b) JPEG2000

Figure 7.18: A comparison between different degrees of pruning.

(a) JPEG2000

(b) Proposed method

Figure 7.19: Lena coded at 0.037bpp with (a) JPEG2000, 1210 bytes and (b) wedgelets with 9 colors, 1201 bytes.

8 Conclusions and Further Directions

If you want to make an apple pie from scratch, you must first create the universe. —Carl Sagan (1934 - 1996).

In this thesis, several novel techniques for image approximation and compression have been presented. Primarily, these proposed techniques focus on improving analysis and synthesis of image edges, which are perceptually important for human vision. In Chapter 1, we propose an image coding framework (see Figures 1.5 and 1.6) that separates image contents into different ordered layers for more efficient analysis and coding. The layer that contains geometrical information such as object edges and shapes have the highest perceptual importance. This concept is similar to how our vision works, in which the retina image is parsed in the primary visual cortex and dozens of other visual areas to differentiate colors, motion, forms and depth information for processing.

Our first contribution concerns embedded coding of color images using wavelets. Pioneering techniques like the *embedded zerotree wavelets* (EZW) and *set partitioning in hierarchical trees* (SPIHT) have been designed for monochrome images only. By exploiting the inter-spectral dependency between wavelet coefficients through a single spatial orientation tree structure, we propose an simple color image coding algorithm, the *color set*

partitioning in hierarchical trees (CSPIHT), that embeds both luminance and chrominance data into a single code stream. CSPIHT offers very comparable reconstruction quality without increasing the complexity order, as compared to the SPIHT+KLT solution. The performance of the CSPIHT scheme is also shown to exceed that of the Color-EZW.

Based on the groundwork of Meyer [85] and Cohen *et al.* [86] for interval wavelets (Type-I), we have designed two new families (Type-II and III) of interval wavelets. These new sets of wavelets, together with that of Cohen *et al.*, allow robust and compact wavelet decomposition of arbitrary finite length sequences without border distortions. In addition, we present a general algorithm to compute the various families of the orthogonal boundary wavelet filters coefficients and their corresponding pre/post-conditioning matrices for any vanishing moments.

Based on interval wavelets, we propose a new expansion for analyzing piecewise smooth signals by avoiding filtering across singularities of interest. In the presence of discontinuities, the interval wavelet is able to retain the same approximation accuracy order as for globally smooth functions. Experiments on non-linear signal approximation shows that interval wavelets can reproduce signals with less artifacts around discontinuities than standard wavelets.

The application of interval wavelets is extended to image compression in order to obtain better edge representation with minimal artifacts. Essentially, each row and column data of the image is analyzed in piecewise manner. However, the extension to 2D signal analysis is not straightforward. There are several problems such as the misalignment of interior filters. Nevertheless, we have presented several algorithms to resolve these issues through the appropriate use of Type-I, II and III interval wavelet filters. Image approximation experiments have demonstrated that, compared to standard wavelet techniques, the proposed 2D interval wavelets decomposition method shows marked visual improvement in the sharpness of image edges.

Interval wavelet transform requires encoding the edge or boundary locations as side information. These boundary information can be represented by objects known as *wedgelets*. Several limitations and inefficiency in image approximations are observed for wedgelet analysis and we introduce a novel *multi-layered wedgelet* technique to overcome

these issues. In the numerical experiments with cel-based cartoon and natural images, our proposed hybrid multiscale wavelet-wedgelet image coding scheme outperforms JPEG2000 [21][22] in terms of visual quality. Additionally, we demonstrate that by using wedgelet representations in very low bit-rate environment, macro features in images can be preserved sufficiently to facilitate visual interpretation.

At the end of this work, we have developed analysis tools that could fulfill some of the functions required for the coding and decoding system proposed in Chapter 1. In the following sections to the end of this thesis, we discuss the limitations of the current works and the possible solutions that could be explored. Additionally, we discuss about some possible areas of future research that could help realize the kind of image coding and decoding system envisioned.

8.1 Interval Wavelets on Short Intervals

For interval wavelet decomposition, we mentioned that there exist a minimum interval length requirement in which existing boundary filters can be applied to the signal. This minimum interval constraint also exists for ENO wavelets and wavelet footprints, which limits the number of decomposition level that could be perform on the interval. Although we suggested using Haar filters for short interval analysis in this work, a more efficient solution is to construct filters that are based on polynomials since it is the objective to generate polynomials up to degree $N-1$ on the short intervals of length N. There exist several methods for constructing polynomial-based interval wavelets such as [83], [**?**] and citeuhlmann2003]. Chebyshev-polynomial wavelets require weights in their scalar products, which can lead to difficulties in interpretation of the relative significance of their coefficients. The Legendre-polynomial wavelets only decays roughly as $x^{?1}$. Nevertheless, there is much potential in designing new approximation bases adapted for intervals, especially those that are based on polynomials.

8.2 Quantization and Coding of Wedgelet Parameters

For application purposes, there is still much room in designing a good coding method for the wedgelet or multi-layered wedgelet parameters. In this work, we show how to prune a wedgelet quad-tree, predict and quantize the wedgelet level constant parameters

and arithmetic code these information to achieve a desired compression ratio. Other parameters are binary uncoded. Due to edge continuity of object outlines, there exist correlation between wedgelet orientation profiles in neighboring partitions. Therefore, further compression can be expected if a comprehensive arithmetic coding scheme can be designed for wedgelets in order to exploit this inter-partition correlation.

8.3 Visual Distortion Measure

In this work, in order to facilitate comparison with experimental results from other publications, approximation results are measured using *Mean-Square-Error* (MSE) *Peak-Signal-Noise-Ratio* (PSNR) criteria. In the image coding and computer vision literature, these are the most commonly used measures for deviations between the original and coded images [123][124][125] due to their mathematical tractability. Moreover, it is straightforward to design coding methods that minimize the MSE, e.g. seeking a optimal wedgelet representation using eqn. (7.5). Since the MSE and PSNR measures account only for pixel-to-pixel differences, they work best when the distortion is due to additive noise. It is important to know that they do not correspond to all aspects of the observer s visual perception of the signal errors [126] nor do they correctly reflect structural coding artifacts [127]. Furthermore, the shortcomings of MSE and PSNR measures are especially obvious at low bit rates when their measured values can contradict observations. Since a human observer is the end user in multimedia applications, an image quality measure that is based on a *Human Vision Model* (HVS) would be more appropriate. However, the HVS is too complex to be fully understood with the current psycho-physical means, but the incorporation of simplified HVS models into quantization models and objective measures exist such as [128][129][130][131][132][133] and [134]. As a future extension of this work, suitable HVS-based error metrics could be incorporated into the analysis and approximation tools to obtain better and more intuitive image reconstruction at low bit rates.

8.4 Texture Synthesis

. The L3 layer in the proposed embedded perceptual framework (see figures 1.5 and 1.6) consists of textures details that is to be added to images constructed from

L2 and L1. Currently, there is no known efficient method for general image texture coding. It is widely accepted that sinusoids and wavelets are not optimal bases for such image features but nevertheless they are currently the only solutions that are simple for the problem. Unlike general image content often assumed for approximation and coding, texture images are spatially homogeneous and noise-like, which is difficult to approximate even with the latest wavelet technology. This stems from the well-known fact that transformation of a noise-like signal tends to yield noise-like output, which does not benefit approximation or coding purposes. Often, texture images contain repetitive structures, often with random variations. This suggests that, instead of coding textures, they could be better reconstructed by generating variations of primitive texture elements at periodic or random spatial positions. Since textures could be noise-like or fractal-like, e.g., grass patch, tree top canopies. reconstruction accuracy for noisy textures is not important in terms of perception. Instead of being transform-coded, it is possible to construct good visual quality textures [135][136][137][138][139] at the decoder using synthetic rendering via a finite set of parameters. Therefore, a suitable coder for the contents from L3 layer would need to adopt techniques for analyzing natural textures and synthetic texture rendering. There have been decades of work on the mentioned subjects especially in the field of computer graphics. It is interesting if these works could be developed further for image coding applications. Indeed, we firmly believe that computer vision and graphics techniques have a big role in the next generation of image and video compression technologies .

9 Publications

- W. S. Lee and A. A. Kassim, *Image Approximation Using Interval Wavelet Transform*, to appear in IEEE Transactions on Image Processing, 2006.

- A. A. Kassim, P. K. Yan, W. S. Lee and K. Sengupta, *Motion Compensated Lossy-to-Lossless Compression of 4-D MRI Data Using Integer Wavelet Transforms*, IEEE Transactions on Information Technology in Biomedicine,, vol. 9, no. 1, pp. 132–138, 2005.

- W. S. Lee and A. A. Kassim, *Issues and Solution Concerning Video Coding Using SPIHT-based schemes*, International Workshop on Advanced Image Technology, Jan. 2004.

- W. S. Lee and A. A. Kassim, *Animation image cel coding using Wedgelets and Beamlets*, Visual Communications and Image Processing, vol. 5150, pp. 1460–1469, June 2003.

- A. A. Kassim and W. S. Lee, *Embedded Color Image Coding Using SPIHT with Partial Linked Spatial Orientation Trees*, IEEE Transactions on Circuits, System & Video Technology, vol. 13, no. 2, pp. 203–206, Feb. 2002.

- A. A. Kassim and W. S. Lee, *Performance of the Color Set Partitioning In Hierarchical*

Tree Scheme (C-SPIHT) in Video Coding, Circuit System Signal Processing, vol. 20, no. 2, pp. 253–270, Oct. 2001.

- W.S. Lee, A.A. Kassim, *Embedded Color Image Coding using Modified Set Partitioning in Hierarchical Tree Scheme*, 6th World Multi Conference on Systemics, Cybernetics & Informatics, vol. XIII, pp 398–403, July 2001.

- W. S. Lee and A. A. Kassim, *Low Bit-rate Video Coding Using Color Set Partitioning In Hierarchical Trees Scheme*, IEEE Inter. Conference on Communication Systems, 2000.

List of References

[1] W. G. Pierpont, *The Art & Skill of Radio-Telegraphy*. Canada: Radio Amateur Educational Society, 1997.

[2] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, July and October 1948.

[3] R. M. Fano, *Transmission of Information*. Cambridge, MA, USA: MIT Press, 1949.

[4] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proceedings of the IRE*, vol. 40, pp. 1098–1101, 1952.

[5] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, pp. 337–342, 1977.

[6] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, pp. 530–536, 1978.

[7] T. A. Welch, "A technique for high-performance data compression," *Computer*, vol. 17, pp. 8–18, June 1984.

[8] I. Witten, R. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Communications of the Association for Computing Machinery*, vol. 30, pp. 520–540, June 1987.

[9] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Tech. Rep. 124, Digital Systems Research Center Research, 1994.

[10] J. B. J. Fourier, *Théorie analytique de la chaleur*. Paris: Firmin Didot, 1882.

[11] J. Peetre, "On Fourier's discovery of Fourier series and Fourier integrals." `http://citeseer.ist.psu.edu/595297.html`.

[12] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Trans. Computer*, vol. C-23, pp. 90–93, Jan 1974.

[13] I. S. G. P. Hudson, H. Yasuda, "The international standardization of a still picture compression technique," in *GLOBECOM '88*, pp. 1016–1021, 1988.

[14] G. K. Wallace, "The JPEG still picture compression standard," *Commin. of the ACM*, vol. 34, pp. 31–44, Apr. 1991.

[15] G. K. W. A. Leger, T. Omachi, "JPEG still picture compression algorithm," *Optical Engineering*, vol. 30, pp. 949–954, July 1991.

[16] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. on Consumer Electronics*, vol. 38, pp. 18–34, Apr. 1992.

[17] W. B. Pennebaker and J. L. Mitchell, *JPEG still image data compression standard*. New York: NY: Van Nostrand Reinhold, 1993.

[18] D. J. L. Gall, "The mpeg video compression algorithm," *Signal Process.: Image Commun.*, vol. 4, pp. 129–140, Apr. 1992.

[19] A. G. MacInnis, "The mpeg systems coding specifications," *Signal Process: Image Commun.*, vol. 4, pp. 153–159, Apr. 1992.

[20] K. R. Rao and J. J. Hwang, *Techniques & Standards for Image Video & Audio Coding*. New Jersey: Prentice Hall PTR, 1996.

[21] "Official JPEG2000 page." `http://www.jpeg.org/jpeg2000/`.

[22] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer, 2001.

[23] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Mathematische Annalen*, vol. 69, pp. 331–371, 1910.

[24] P. Levy, "Theorie de l'addition des variables aleatoires," *Paris : Gauthier-Villars*, 1937.

[25] P. Levy, "Le mouvement Brownien plan," *American Journal of Mathematics*, vol. 62, no. 1/4, pp. 487–550, 1940.

[26] P. Levy, "Processus stochastiques et mouvement Brownien," *Paris : Gauthier-Villars*, 1948.

[27] A. Grossman and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape," *Society for Industrial and Applied Mathematics Journal on Mathematical Analysis*, no. 15, pp. 732–736, 1984.

[28] B. Burke, "The mathematical microscope: Waves, wavelets, and beyond," in *A Positron Named Priscilla: Scientific Discovery at the Frontier*, pp. 196–235, National Academy of Sciences, 1994.

[29] Y. Meyer, "Ondelettes et fonctions splines," in *Séminaire sur les équations aux dérivées partielles 1986–1987*, pp. Exp. No. VI, 18, Palaiseau: École Polytech., 1987.

[30] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 11, pp. 674–693, Jul 1989.

[31] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Commun. on Pure and Appl. Math.*, vol. 41, pp. 909–996, Nov 1988.

[32] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," *IEEE Transactions on Information Theory*, vol. 36, pp. 961–1005, Sept. 1990.

[33] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.

[34] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Techn.*, vol. 6, pp. 243–250, 1996.

[35] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on Image Processing*, vol. 9, pp. 1158–1170, July 2000.

[36] T. Nørretranders, *Merk Verden: en Beretning om Bevissthet*. Oslo, Norway: Cappelens Forlag, 1992.

[37] D. L. Donoho, "Wedgelets: Nearly-minimax estimation of edges," *Annals. of Stat.*, vol. 27, pp. 859–897, 1999.

[38] E. L. Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE trans. on Image Processing*, vol. 14, pp. 423–438, April 2005.

[39] E. J. Candès and D. L. Donoho, "Curvelets, multiresolution representation, and scaling laws," in *Wavelet Appl. in Signal and Image Processing VIII* (A. Aldroubi, A. F. Laine, and M. A. Unser, eds.), vol. 4119, SPIE Press, 2000.

[40] M. N. Do and M. Vetterli, "Contourlets: A directional multiresolution image representation," in *Proc. of IEEE Inter. Conf. on Image Processing (ICIP)*, (Rochester (USA)), Sept 2002.

[41] M. Do and M.Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE trans. on Image Processing*, vol. 14, pp. 2091–2106, December 2005.

[42] E. J. Candès and D. L. Donoho, "Ridgelets: a key to higher dimensional intermittency?," *Philos. Trans. Roy. Soc. London Ser.*, vol. 357, pp. 2495–2509, Sept 1999.

[43] L. Duval, "Where is the Starlet? X-lets." `http://lcd.siva.free.fr/where_is_the_starlet.html#xlet`.

[44] J. Baudrillard, *Simulacra and Simulation*. Ann Arbor, MI, USA: University of Michigan Press, 1995.

[45] T. Szirányi and Z. Tóth, "Random paintbrush transformation," in *15th ICPR*, (Barcelona), pp. 155–158, IAPR & IEEE, 2000.

[46] W. K. Pratt, *Digital image processing*. John Wiley, 2nd ed., 1991.

[47] J. Bertin, *The Semiology of Graphics*. University of Wisconsin Press, 1983.

[48] H. von Helmholtz, "Concerning the perceptions in general," *Treatise on physiological optics*, vol. 3, pp. 1–37, 1866.

[49] H. von Helmholtz, "Concerning the perceptions in general (translated)," *Optical Society of America*, 1924.

[50] S. E. Palmer, *Vision science: Photons to Phenomenology*. Cambridge, MA: MIT Press, 1999.

[51] J. Bertin, *Sémiologie Graphique: Les diagrammes, les réseaux, les cartes*. Gauthier-Villars, 1967.

[52] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *Journal Physiol.*, vol. 148, pp. 574–591, 1959.

[53] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal Physiol.*, vol. 160, pp. 106–154, 1962.

[54] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture in two non-striate visual areas (18 and 19) of the cat," *Journal Neurophysiol.*, vol. 28, pp. 229–289, 1965.

[55] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *Journal Physiol.*, vol. 195, pp. 215–243, 1968.

[56] K. Koffka, *Principles of Gestalt Psychology*. NY: Harcourt, Bruce and Company, 1935.

[57] E. G. Boring, *Sensation and perception in the history of experimental psychology*. Appleton-Century-Crofts, 1942.

[58] A. G. Leventhal, Y. C. Wang, M. T. Schmolesky, and Y. Zhou, "Neural correlates of boundary perception," *Visual Neuroscience*, vol. 15, pp. 1107–1118, Nov 1998.

[59] R. Arnheim, *Art and Visual Perception*. Faber & Faber, 1968.

[60] A. L. Yarbus, *Eye Movements and Vision*. Plenum Press, 1967.

[61] M. A. Fischler and O. Firschein, *Intelligence: The Eye, the Brain, and the Computer*. Addison-Wesley, 1987.

[62] D. Forsyth and J. Ponce, *Computer Vision A Modern Approach*. Prentice Hall, 2002.

[63] P. H. Winston, *Artificial Intelligence*. Addison-Wesley, 1992.

[64] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *16th International Conference of Pattern Recognition*, pp. 150–155, 2002.

[65] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, pp. 532–540, April 1984.

[66] S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $l^2(r)$," *Trans. of American Mathematical Soc.*, vol. 315, no. 7, pp. 69–87, 1989.

[67] Y. Meyer, *Ondellettes et opératuers*. Paris: Hermann, 1990.

[68] Y. Meyer, *Wavelets and Operators*. Cambridge University Press, 1992.

[69] I. Daubechies and J. C. Lagarias, "Two-scale difference equations II. Local regularity, infinite products of matrices and fractals," *SIAM J. Math. Anal.*, vol. 23, no. 4, pp. 1031–1079, 1992.

[70] I. Daubechies and J. C. Lagarias, "Two-scale difference equations I. Existence and global regularity of solutions," *SIAM J. Math. Anal.*, vol. 22, no. 5, pp. 1388–1410, 1991.

[71] I. Daubechies, *Ten lectures on wavelets*. Philadelphia: SIAM, 1992.

[72] C. Herley and M. Vetterli, "Wavelets and recursive filter banks," *IEEE Trans. Signal Processing*, vol. 48, no. 8, pp. 2536–2556, 1993.

[73] S. Mallat, *A Wavelet Tour of Signal Processing*, ch. 6, pp. 176–188. Academic Press, 2nd ed., 1999.

[74] V. R. Algazi and R. R. Estes, "Analysis based coding of image transform and subband coefficients," *Proceedings of the SPIE*, vol. 25, no. 64, pp. 11–21, 1995.

[75] A. A. Liff, *Color and black and white: television theory and servicing*. Prentice Hall, 1993.

[76] A. N. Netravali and B. G. Haskell, *Digital pictures: representation and compression, 3rd edition*. Applications of Communication Theory, Plenum Press, 1988.

[77] A. Said and W. A. Pearlman, "SPIHT FAQ: What method is used for color compression?." `http://www.cipr.rpi.edu/research/SPIHT/spiht6.html`, 2002.

[78] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *The Journal of Educational Psychology*, vol. 8, pp. 419–448, 1933.

[79] K. Shen and E. J. Delp, "Color image compression using an embedded rate scalable approach," in *IEEE Inter. Conf. On Image Processing*, (Santa Barbara, California), pp. III34–III37, Oct. 1997.

[80] T. F. Chan and H. M. Zhou, "CAM report: Adaptive ENO-wavelet transforms for discontinuous functions," Tech. Rep. 99-21, UCLA, June 1999.

[81] J. R. Williams and K. Amaratunga, "A discrete wavelet transform without edge effects using wavelet extrapolation," *Journal of Fourier Analysis and Applications*, vol. 3, no. 4, pp. 435–449, 1997.

[82] C. Herley, "Boundary filters for finite-length signals and time-varying filter banks," *IEEE International Symposium on Circuits and System (ISCAS)*, vol. 27, pp. 637–640, May 1994.

[83] T. Kilgore and J. Prestin, "Polynomial wavelets on the interval," *Constr. Approx.*, vol. 12, no. 1, pp. 95–110, 1994.

[84] J. Fr ohlich and M. Uhlmann, "Orthonormal polynomial wavelets on the interval and applications to the analysis of turbulent flow fields," *SIAM J. Appl. Math.*, vol. 63, no. 5, pp. 1789–1830, 2003.

[85] Y. Meyer, "Ondelettes sur l'intervalle," *Rev. Mat. Iberoamericana*, vol. 71, no. 2, pp. 115–133, 1991.

[86] A. Cohen, I. Daubechies, and P. Vial, "Wavelets on the interval and fast wavelet transforms," *Journal of Appl. Comput. Harmon. Anal.*, vol. 1, no. 1, pp. 54–81, 1994.

[87] F. Chyzak, P. Paule, O. Scherzer, A. Schoisswohl, and B. Zimmermann, "The construction of orthonormal wavelets using symbolic methods and a matrix analytical approach for wavelets on the interval," *Experimental Mathematics*, vol. 10, no. 124, pp. 66–86, 2001.

[88] G. Strang and G. Fix, "A Fourier analysis of the finite element variational method," *Constructive Aspects of Functional Analysis*, pp. 796–830, June 1971.

[89] F. Chaplais, "Algebras and nonlinear multiresolution analysis that are consistent with the Strang and Fix conditions," in *Proc. of the IEEE-SP Inter. Symp. on Time-Frequency and Time-Scale Analysis*, pp. 445–448, June 1996.

[90] P. G. Lemarié and G. Malgouyres, "Support des fonctions de base dans une analyse multirésolution," *C. R. Acad. Sci. Paris*, no. 313, pp. 377–380, 1991.

[91] A.Cohen, I. Daubechies, and J.C.Feauveau, "Biorthogonal bases of compactly supported wavelets," *Comm Pure and Applied Math,*, vol. 45, pp. 485–560, 1992.

[92] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges," *IEEE Trans. Patt. Recog. and Mach. Intell.*, vol. 14, pp. 710–732, July 1992.

[93] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," *IEEE Trans. on Signal Proc.*, vol. 38, pp. 617–643, March 1992.

[94] P. L. Dragotti and M. Vetterli, "Footprints and edgeprints for image denoising and compression," in *Proc. of IEEE Inter. Conf. on Image Processing (ICIP)*, pp. 237–240, October 2001.

[95] P. L. Dragotti and M. Vetterli, "Wavelet footprints: Theory, algorithms and applications," *IEEE Trans. on Signal Processing*, vol. 51, pp. 1306–1323, May 2000.

[96] S. Osher, A. Harten, B. Engquist, and S. Chakravarthy, "Uniformly high order essentially non-oscillatory schemes III," *Journal of Computational Physics*, vol. 71, pp. 231–303, 1987.

[97] S. Amat, F. Aràndiga, A. Cohen, R. Donat, G. Garcia, and M. von Oehsen, "Data compression with ENO schemes," Tech. Rep. 99-03, Universitat de València, Sept 1999.

[98] T. F. Chan and H. M. Zhou, "Adaptive ENO-wavelet transforms for discontinuous functions," in *Proc. of the 12th Inter. Conf. on Domain Decomposition Methods*, (Chiba (Japan)), 2001.

[99] T. F. Chan and H. M. Zhou, "ENO-wavelet transforms and some applications," in *Beyond Wavelets*, pp. 1–34, Academic Press, 2001.

[100] K. Amaratunga and J. R. Williams, "Time integration using wavelets," in *Proceedings of SPIE: Wavelet Applications for Dual Use*, vol. 2491, pp. 894–902, Apr. 1995.

[101] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley, MA, USA: Wellesley-Cambridge Press, 1995.

[102] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679–698, 1986.

[103] M. Hueckel, "An operator which locates edges in digital pictures," *J. ACM*, vol. 18, no. 1, pp. 113–125, 1971.

[104] L. Mero, "A simplified and fast version of the hueckel operator for finding optimal edges in pictures," *Pric. IJCAI*, vol. 37, pp. 650–655, 1975.

[105] R. Nevatia, "Evaluation of simplified hueckel edgeline detector," *Comput. Graph. Image Process.*, vol. 6, no. 6, pp. 582–588, 1977.

[106] L. Carpenter, "The A-buffer, an antialiased hidden surface," *Sigraph*, pp. 103–108, 1984.

[107] F. C. Crow, "A comparison of antialiasing techniques," *IEEE Computer Graphics and Applications*, vol. 1, pp. 40–48, Jan. 1981.

[108] N. L. Max, "Antialiasing scan-line data," *IEEE Computer Graphics and Applications*, vol. 10, pp. 18–30, January 1990.

[109] X. Huo and D. L. Donoho, "Beamlets and multiscale image analysis," in *Lecture Notes in Computational Science and Eng.: Multiscale and Multiresolution methods*, pp. 149–195, Springer, 1999.

[110] W. S. Lee and A. A. Kassim, "Animation image cel coding using wedgelets and beamlets," in *Visual Comm. and Image Processing 2003* (T. Ebrahimi and T. Sikora, eds.), vol. 5150, pp. 1460–1469, Lugano (Switzerland): SPIE Press, June 2003.

[111] M. L. Brady, "A fast discrete approximation algorithm for the radon transform," *SIAM Journal of Comput.*, vol. 27, pp. 107–119, February 1998.

[112] D. L. Donoho, "A fast discrete approximation algorithm for the radon transform," *SIAM Journal on Mathematical Analysis*, vol. 31, no. 5, pp. 1062–1099, 2000.

[113] H. Führ, "Efficient implementation of wedgelet approximations," *HASSIP Workshop in Cambridge*, Sept. 2004.

[114] J. Romberg, M. Wakin, and R. Baraniuk, "Multiscale wedgelet image analysis: fast decompositions and modeling," in *Proc. of IEEE Inter. Conf. on Image Processing (ICIP)*, pp. 585–588, Sept. 2002.

[115] L. Demaret, F. Friedrich, H. Führ, and K. Wicker, "Discrete Green's theorem for polygonal domains, with an application to rapid wedgelet approximation (preprint 2005)." `http://ibb.gsf.de/preprints.php`, 2005.

[116] A. Zaccarin and B. Liu, "Transform coding of color images with limited palette size," *Pro. ICASSP*, pp. 2625–2628, May 1991.

[117] A. Zaccarin and B. Liu, "A novel approach for coding color quantized images," *IEEE Trans. Image Proc.*, vol. 2, pp. 2625–2628, October 1993.

[118] W. Kim and R. Park, "Color image palette construction based on the HSI color system for minimizing the reconstruction error," *ICIP '96*, pp. 1041–1044, 1996.

[119] P. Waldemar and T. Ramstad, "Subband coding of color images with limited palette size," *Pro. ICASSP*, pp. 353–356, May 1994.

[120] D. Youla, "Mathematical theory of image restoration by the method of convex projections," in *Image Recovery: Theory and Applications* (H. Stark, ed.), Academic Press, 1987.

[121] Y. Liu, "A POCS-based representation algorithm for arbitrarily shaped image segments," *SCI2001*, vol. VI, July 2001.

[122] W. S. Lee and A. A. Kassim, "Embedded color image coding using SPIHT with partially linked spatial orientation trees," *IEEE Trans. on Circuits and Systems for Video Techn.*, vol. 13, no. 2, pp. 203–206, 2003.

[123] H. de Ridder, "Minkowsky metrics as a combination rule for digital image coding impairments," in *Proceedings SPIE: Human Vision, Visual Processing and Digital Display III*, vol. 1666, pp. 17–27, 1992.

[124] A. M. E. lu and P. S. Fisher, "Image quality measures and their performance," *IEEE Trans. Commun.*, vol. 43, no. 12, pp. 2959–2965, 1995.

[125] A. M. Eskicioğlu, "Application of multidimensional quality measures to reconstructed medical images," *Opt. Eng.*, vol. 35, no. 3, pp. 778–785, 1996.

[126] B. Girod, "Whats wrong with mean-squared error," in *Digital Images and Human Vision*, pp. 207–220, Cambridge, MA: MIT Press, 1993.

[127] S. Daly, "The visible differences predictor: An algorithm for the assessment of image fidelity," in *Digital Images and Human Vision*, pp. 179–205, Cambridge, MA: MIT Press, 1993.

[128] N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Trans. Commun.*, vol. 33, no. 6, pp. 551–557, 1985.

[129] N. B. Nill and B. H. Bouzas, "Objective image quality measure derived from digital image power spectra," *Opt. Eng.*, vol. 31, no. 4, pp. 813–825, 1992.

[130] T. Frese, C. A. Bouman, and J. P. Allebach, "Methodology for designing image similarity metrics based on human visual system models," in *Proc. SPIE IS&T Conf. on Human Vision and Electronic Imaging II*, vol. 3016, pp. 472–483, 1997.

[131] J. Chen and T. Pappas, "Perceptual metrics and perceptual coders," in *Proceedings SPIE: Human Vision and Electronic Imaging*, vol. 4299, Jan. 2001.

[132] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, pp. 81–84, Mar. 2002.

[133] Z. Wang, A. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Proc.*, vol. 13, no. 4, pp. 600–612, 2004.

[134] W. S. Lin, L. Dong, and P. Xue, "Visual distortion gauge based on discrimination of noticeable contrast changes," *IEEE Trans. on Circuits and Systems for Video Techn.*, vol. 15, pp. 900–909, July 2005.

[135] D. Cano and T. H. Minh, "Texture synthesis using hierarchical linear transforms," *Signal Processing*, vol. 15, pp. 131–148, 1988.

[136] M. Porat and Y. Y. Zeevi, "Localized texture processing in vision: Analysis and synthesis in gaborian space," *IEEE Trans. Biomedical Eng.*, vol. 36, pp. 115–129, Oct 1989.

[137] D. Heeger and J. Bergen, "Pyramid-based texture analysis/synthesis," in *ACM SIGGRAPH*, August 1995.

[138] J. Portilla, R. Navarro, O. Nestares, and A. Tabernero, "Texture synthesis-by-analysis based on a multiscale early-vision model," *Optical Engineering*, vol. 35, no. 8, pp. 2403–2417, 1996.

[139] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int'l Journal of Computer Vision*, vol. 40, pp. 49–71, Oct 2000.